УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Данијела Симић

ФОРМАЛИЗАЦИЈА РАЗЛИЧИТИХ МОДЕЛА ГЕОМЕТРИЈЕ И ПРИМЕНЕ У ВЕРИФИКАЦИЈИ АУТОМАТСКИХ ДОКАЗИВАЧА ТЕОРЕМА

докторска дисертација

Београд, 2017.

UNIVERSITY OF BELGRADE
FACULTY OF MATHEMATICS

Danijela Simic

FORMALIZATION OF VARIOUS GEOMETRY MODELS AND APPLICATIONS IN VERIFICATION OF AUTOMATED THEOREM PROVERS

Doctoral Dissertation

Belgrade, 2017.

Ментор:

др Филип Марић, ванредни професор
Универзитет у Београду, Математички факултет

Чланови комисије:

др Предраг Јаничић, редовни професор
Универзитет у Београду, Математички факултет

др Срђан Вукмировић, ванредни професор
Универзитет у Београду, Математички факултет

др Петар Максимовић, Research Fellow
Imperial College London
Научни сарадник, Математички институт САНУ

Датум одбране: _____

родитељима, Милијани и Драгану Петровићу

Захвалница

Велику захвалност дугујем свом ментору, професору Филипу Марићу. Професоров велики ентузијазам, знање и безрезервна подршка су ми много помогли у току докторских студија и приликом рада на различитим истраживањима и пројектима. Он ме је упутио у диван свет интерактивног доказивања теорема и бројним саветима и сугестијама ме је мотивисао да што више научим и напредујем. Посебно истичем његове иновативне идеје и позитивност која је често утицала да са еланом и новом енергијом наставим да се бавим проблемима на које сам у раду наишла. Захваљујући њему сам објавила радове у часописима и конференцијама и завршила ову тезу, а без његове помоћи и подршке тога не би било.

Захваљујем се и професору Предрагу Јаничићу, он ме је упутио у свет геометрије и аутоматског доказивања у геометрији. Многим саветима и сугестијама је утицао да финална верзија тезе буде квалитетнија и потпунија. Захваљујући њему упознала сам и многе истраживаче из других земаља и тако ми је створена могућност да размењујем искуства и да учествујем у пројектима и ван граница наше земље. Напоменула бих и да са професором сарађујем у настави и да је вишегодишње заједничко држање курсева било једно лепо и пријатно искуство.

Захваљујем се и професорима Срђану Вукмировићу и Петру Максимовићу на пажљивом читању тезе. Њихови коментари су значајно унапредили текст тезе, али су дали и нови поглед на проблеме којима се бавим и пружили су ми нове идеје како да истраживање наставим.

Захваљујем се и свим члановима катедре, професорима и асистентима. Сви су били заиста дивни сарадници и од првих дана рада ми значајно помогли да напредујем као наставник и као истраживач. Дали су ми бројне савете, одменили када је било потребно или улепшали дан на послу. Могу рећи да сам заиста срећна што радим у тако динамичном, позитивном и колегијалном колективу.

Посебну захвалност дугујем својим родитељима којима ову тезу посвећујем. Они су ми били ослонац и подршка током целог живота. Захваљујући њиховој неизмерној љубави и упорности сам остварила многе циљеве на пословном и приватном плану. Од првог дана школовања су веровали у мене и

бодрили ме, помагали и упућивали да увек тежим да више научим, сазнам и постигнем.

Коначно, захваљујем се и свом супругу Михаилу. Био је велика подршка у раду на овој тези, одмењивао ме у разним пословима, подизао дух када бих посустала, више пута заједно са мном читао тезу и увек веровао да ћу успети. Његова љубав и позитивна енергија су ми учинили рад на тези лакшим и лепшим.

Наслов дисертације: Формализација различитих модела геометрије и примене у верификацији аутоматских доказивача теорема

Резиме:

У овој тези представљена је интерактивна формализација модела разних геометрија и алгебарских метода аутоматског доказивања геометријских теорема.

Представљен је наш рад на формализацији аналитичке (Декартове) планарне геометрије у оквиру асистента за доказивање теорема *Isabelle/HOL*. Дајемо неколико еквивалентних дефиниција Декартове координатне равни и доказујемо да је она модел синтетичких планарних геометрија (коришћењем аксиоматског система Тарског и аксиоматског система Хилберта). Такође, дискутујемо о неколико техника којима се поједностављују и аутоматизују докази. Како је један од наших циљева да подржимо коришћење асистената за доказивање теорема у математичком образовању, наше излагање ће бити блиско стандардним дефиницијама у уџбеницима, али потпуно формално и машински провериво. Ова формализација представља део потребне инфраструктуре за имплементацију процедура одлучивања базираних на аналитичкој геометрији у оквиру асистената за доказивање теорема.

Додатно, формално разматрамо и геометрију комплексне равни. Блиска повезаност између комплексних бројева и геометрије је добро позната и пажљиво је изучавана још вековима уназад. Основни објекти који су изучавани су комплексна раван (обично проширена једном бесконачном тачком), њени објекти (тачке, праве и кругови), и група трансформација која на њих делује (на пример, инверзије и Мебијусове трансформације). У овој тези, ми формално посматрамо геометрију комплексних бројева и представљамо потпуно механички верификовану теорију у оквиру асистента за доказивање теорема *Isabelle/HOL*. Дискутујемо о различитим приступима формализацији и главним предностима приступа који је више алгебарски оријентисан. Поред примена у формализацији математике и у образовању, овај рад је основа за формалну анализу нееуклидских геометрија и њихове међусобне повезаности. Такође, представљамо и формализацију дела аксиоматског система Тарског у оквиру Поенкареовог диск модела у систему *Isabelle/HOL*.

Треће, анализирамо везу између геометрије и полинома, као и примене која ова веза даје. У еуклидској геометрији објекти и релације међу њима

могу се изразити полиномијалним једнакостима. Додатно, било која геометријска конструкција може се изразити скупом полиномијалних једнакости, а геометријска тврђења се могу доказати коришћењем алгебарских метода (на пример, метод Гребнерових база или Вуовом методом) над скупом полинома. Дајемо опис алгоритма у систему *Isabelle/HOL* који као улазни податак прихвата геометријску конструкцију записану коришћењем термова, а враћа одговарајући скуп полинома. Наш даљи рад ће бити примена методе Гребнерових база у оквиру система *Isabelle/HOL* над генерисаним полиномима у намери да се докаже исправност дате конструкције. Додатно, истражујемо како се конструкције у тродимензионалном простору могу приказати коришћењем полинома. Истражујемо два различита приступа у извођењу ових полинома и онда поредимо ефикасност алгебарских метода у зависности од коришћеног приступа. Представљамо потпуно аутоматски систем за превођење геометријских конструкција из стереометрије у скуп полинома. Наш даљи рад ће бити да повежемо представљени систем са динамичким геометријским софтвером и на тај начин да омогућимо студентима лакше коришћење овог аутоматизованог система за доказивање у стереометрији.

Кључне речи: асистент за доказивање теорема, геометрија, интерактивно доказивање у геометрији, аутоматско доказивање у геометрији, хиперболичка геометрија, стереометрија, аксиоматски систем Тарског, аксиоматски систем Хилберта, Декартов координатни систем

Научна област: рачунарство

Ужа научна област: аутоматско резоновање

УДК број: 004.832.3:514(043.3)

Dissertation title: Formalization of various geometry models and applications in verification of automated theorem provers

Abstract:

In this thesis is presented interactive formalization of various models of geometry and algebraic methods for automated proving geometry theorems.

We present our current work on formalizing analytic (Cartesian) plane geometries within the proof assistant Isabelle/HOL. We give several equivalent definitions of the Cartesian plane and show that it models synthetic plane geometries (using both Tarski's and Hilbert's axiom systems). We also discuss several techniques used to simplify and automate the proofs. As one of our aims is to advocate the use of proof assistants in mathematical education, our exposure tries to remain simple and close to standard textbook definitions, but completely formal and mechanically verifiable. This formalization presents the develop of the necessary infrastructure for implementing decision procedures based on analytic geometry within proof assistants.

Furthermore, we investigate complex numbers. Deep connections between complex numbers and geometry had been well known and carefully studied centuries ago. Fundamental objects that are investigated are the complex plane (usually extended by a single infinite point), its objects (points, lines and circles), and groups of transformations that act on them (e.g., inversions and Möbius transformations). In this thesis we treat the geometry of complex numbers formally and present a fully mechanically verified development within the theorem prover Isabelle/HOL. We discuss different approaches to formalization and discuss major advantages of the more algebraically oriented approach. Apart from applications in formalizing mathematics and in education, this work serves as a ground for formally investigating various non-Euclidean geometries and their intimate connections. We also present a formalization of part of Tarski axiom system withing Poincare disk model in Isabelle/HOL.

Further on, we analyze connections between geometry and polynomials and the use of these connections. In Euclidean geometry, objects and relations between them can be expressed as polynomials. Further, any geometry construction can be expressed by set of polynomials and geometry statements can be proved by using algebraic methods (e.g. the Gröbner bases method or Wu's method) over that set of polynomials. We describe an implementation of an algorithm in Isabelle/HOL

that accepts a term representation of a geometry construction and returns its corresponding set of polynomials. Our further work will be to use the method of Gröbner bases within the Isabelle system on the generated polynomials, in order to prove correctness of the given construction. Furthermore, we investigate how spatial geometry constructions can be presented using polynomials. We investigate two different approaches in deriving those polynomials and then compare efficiency of algebraic provers depending on the approach used. We present a fully automated system for transforming geometry constructions into set of polynomials. Our further work would be to relate these geometry provers with dynamic geometry software and thus make easier for students to use it.

Keywords: proof assistants, geometry, interactive proving in geometry, automated proving in geometry, hyperbolic geometry, spatial geometry, Tarski axiom system, Hilbert axiom system, Cartesian coordinate system

Research area: computer science

Research sub-area: automated reasoning

UDC number: 004.832.3:514(043.3)

Садржај

Глава 1

Увод

1.1 Мотивација и циљ тезе

У класичној математици постоји много различитих геометријских теорија. Такође, различита су и гледишта шта се сматра стандардном (еуклидском) геометријом. Понекад, геометрија се дефинише као независна формална теорија, а понекад као специфични модел. Наравно, везе између различитих заснивања геометрије су јаке. На пример, може се доказати да Декартова раван представља модел формалних аксиоматских теорија еуклидске геометрије.

Традиционална еуклидска (синтетичка) геометрија је још од античке Грчке заснована на често малом скупу основних појмова (на пример, тачке, праве, основних геометријских релација попут инциденције, подударности итд.) и на скупу аксиома које имплицитно дефинишу ове основне појмове. Иако су Еуклидови „Елементи" један од најутицајних радова из математике, поставило се озбиљно питање да ли је систем аксиома, теорема и лема којима се геометрија описује заиста прецизан. Испоставило се да су нађене грешке у доказима, а и да су неки докази били непотпуни јер су имали имплицитне претпоставке настале због погрешне интуиције или погрешног позивања на слике (дијаграме). Ове празнине су утицале на појаву других аксиоматских система чији је циљ био да дају формалну, прецизнију аксиоматизацију Еуклидове геометрије. Најважнији од тих система су Хилбертов систем аксиома и систем аксиома Тарског.

Хилбертов систем уводи три основна појма (тачка, права и раван), 6 релација и 20 аксиома подељених по групама. Хилберт је желео да направи

систем који је прецизнији од Еуклидовог, у којем ништа није остављено интуицији. Овакав приступ је повећао ниво ригорозности не само у геометрији, него у другим областима математике.

Систем Тарског је мањи, уводи један основни појам (тачка), 2 релације и 11 аксиома. Његова основна предност у односу на Хилбертов систем је у његовој једноставности. Са друге стране, систем Тарског уводи појам праве као скупа тачака што доста отежава резоновање јер захтева да се у доказима користи теорија скупова.

Једно од најзначајнијих открића у математици, које датира из XVII века, јесте Декартово откриће координатног система, које је омогућило да се алгебарским изразима представе геометријске фигуре. То је довело до рада на новој математичкој области која је названа *аналитичка геометрија*.

У математичком образовању у средњим школама и на факултетима често се демонстрирају оба приступа у геометрији (аналитички и синтетички). Ипак, док се синтетички приступ предаје као ригорозан систем (са намером да се демонстрира формалан, аксиоматски приступ изградње математичких теорија), аналитичка геометрија се показује много мање формално. Такође, ова два приступа се уводе независно и веза између њих се ретко формално показује у оквиру стандардног наставног плана.

Иако се појам сферне геометрије појавио још у старој Грчкој, озбиљније истраживање нееуклидских геометрија (сферне, хиперболичке и др.) је започето 1829. године са радом Лобачевског. Ипак, са њиховим интензивнијим истраживањем се почело тек пола века касније. Оно што је највише утицало на ову промену јесте откриће комплексних бројева крајем XVIII века. Комплексни бројеви су представљали значајну алатку за истраживање особина објеката у различитим геометријама. Заменом Декартове координатне равни комплексном равни добијају се једноставније формуле које описују геометријске објекте. Након Гаусове теорије о закривљеним површинама и Римановог рада о многострукостима, геометрија Лобачевског добија на значају. Ипак, највећи утицај има рад Белтрамија који доказује да дводимензионална нееуклидска геометрија није ништа друго до геометрија неке површи константне негативне кривине. Хиперболичка геометрија се изучава кроз многе њене моделе. Уводи се појам пројективног диск модела који Клајн касније популаризује. Поенкаре посматра полуравански модел који су предложили Лиувил и Белтрами и пре свега изучава изометрије хиперболичке равни које чувају

оријентацију. Данас се те трансформације и у ширем контексту изучавају у оквиру Мебијусових трансформација.

Потреба за ригорозним заснивањем математике постоји веома дуго и са развојем математике повећавао се и степен ригорозности. Међу наукама, математика се издваја својим прецизним језиком и јасним правилима аргументовања, тј. извођења. Ова чињеница омогућава да се тачност математичких тврђења аргументују формалним извођењима, тј. доказима. Још у седамнаестом веку, постојала је идеја да мора постајати неки општи језик којим би се могла записати математичка тврђења и општи систем правила за извођење. Један од најзначајних напредака у математици почетком двадесетог века било је откриће да се математички аргументи могу представити у формалним аксиоматским системима на такав начин да се њихова исправност може једноставно испитати коришћењем једноставних механичких правила. Генерално, математика се могла формализовати коришћењем аксиоматске теорије скупова, теорије типова, логике вишег реда и слично. Математички доказ је ригорозан ако може бити записан у некој формалној логици као низ закључака који су изведени применом јасно дефинисаних правила.

Често, механички проверени докази попуњавају празнине које постоје у дефиницијама и доказима и упућују на дубљу анализу теме која се изучава. У историји математике постоји пуно контроверзи око исправности математичких доказа. Године 1935. Лекат је објавио књигу о грешкама које су до 1900. године направили познати математичари. Поред грешака, често се дешавало да математичари нису умели да одреде да ли је неки доказ исправан или не и дешавало се да се у потпуности верује да је доказ тачан ако га је објавио познати математичар, као Гаус или Коши, и њихови докази нису подлегали дубљој критици. У деветнаестом веку докази постају све комплекснији и математичари почињу да све више истичу важност ригорозности доказа. Математичари се свакодневно сусрећу са прескоченим корацима у доказима, са непрецизним дефиницијама, са хипотезама и претпоставкама које недостају. Понекад грешке у доказима не буду примећене веома дуго. На пример, први доказ теореме о обојивости графа са четири боје је имао грешку која је уочена тек десет година касније. Иако је грешке углавном лако исправити, има случајева када је то веома тешко. На пример, 1980. године објављено је да је завршена класификација простих коначних група, али је примећено да постоји пропуст у једној од класа и исправка тог пропуста објављена је тек

2001. године, а доказ је имао 1221 страну [4]. Додатно, често се дешава да се одређени делови доказа никада не прикажу, често уз реченицу „специјалан случај се тривијално доказује" при чему се дешава да за тај специјалан случај тврђење не важи или га није тривијално доказати. Поред овога, понекад је потребно много времена да би се неки доказ проверио. На пример, доказ Кеплерове хипотезе коју је саставио Томас Хејлс има 300 страна и 12 рецензената су провели четири године у анализи доказа и коначно су написали да су 99% сигурни да је доказ исправан.

Многи научници су сматрали да је потпуна формализација математике недостижни идеал. Са појавом рачунара настала је могућност генерисања машински проверљивих доказа. Тако су се појавили системи за формално доказивање теорема. Постоје системи који омогућавају потпуно аутоматску конструкцију доказа и они користе технике попут SAT решавача, технике презаписивања, резолуцију, алгебарске доказиваче. Иако је изградња система за потпуно аутоматско доказивање теорема важан подухват, постоје, за сада, мале реалне могућности да се направи систем који заиста аутоматски доказује компликована математичка тврђења.

Зато је посебан акценат на системима који се заснивају на интеракцији корисника и рачунара. Такви системи су полуаутоматски и у процесу формалног доказивања теорема од стране корисника (често информатичар и/или математичар) помажу тако што контролишу исправност доказа и, колико је то могуће, проналазе аутоматске доказе. Ови *интерактивни доказивачи* се називају и *асистенти за доказивање теорема*. Данас постоји мноштво интерактивних доказивача: *Isabelle*, *Isabelle/HOL*, *Coq*, *HOL Light*, *PVS* и други. Посебно се истичу *Isabelle/HOL* и *Coq* као системи са великим бројем корисника који су током година развили велики скуп библиотека са формално доказаним теоријама које је могуће даље надограђивати. Асистенти за доказивање теорема се користе у различитим областима. Пре свега могу се користити за формалну верификацију рачунарских програма. Поред тога, значајна примена је и у образовању. Помажу развој и продубљивање математичког знања.

Интересовање за аутоматско доказивање у геометрији постоји још одавно. Један од првих аутоматских доказивача теорема уопште био је аутоматски доказивач за геометрију. Тарски је развио алгебарску методу за доказивање теорема еуклидске геометрије, али је она била неупотребљива за компликова-

не теореме. Највећи напредак је направљен тек средином двадесетог века када је Ву предложио своју алгебарску методу за доказивање теорема у еуклидској геометрији. Његовом методом могле су се доказати и веома комплексне теореме. Још једна алгебарска метода која се развила у исто време је метода Гребнерових база. Ови методи имају алгебарски, тј. аналитички приступ у доказивању и заснивају се на репрезентацији тачака коришћењем координата. Модерни доказивачи теорема који се заснивају на овим методама могу да докажу стотине нетривијалних теорема. Ипак, велика мана ових система је што не производе класичне доказе, већ само пропратне аргументе који нису читљиви. Деведесетих година XX-ог века постојало је више покушаја да се овај проблем реши и развијене су нове методе засноване на аксиоматизацији синтетичке геометрије – метода површина, метода пуног угла, итд. Ипак, њихова главна мана је што су далеко мање ефикасни у односу на алгебарске методе. Већина система са аналитичким приступом за доказивање теорема се користи као софтвер којем се верује иако нису формално верификовани. Да би се повећала њихова поузданост потребно их је повезати са модерним интерактивним доказивачима теорема и то је могуће учинити на два начина – њиховом имплементацијом у оквиру интерактивног доказивача теорема и доказивањем њихове исправности или коришћењем интерактивних доказивача да провере њихове сертификате. Неколико корака у овом правцу је већ направљено [53, 113].

Примена система за аутоматско доказивање теорема у геометрији је велика. На пример могу се користити у образовању. Поред тога, користе се у научним областима као што су роботика, биологија, препознавање слика и другим.

1.2 Доприноси тезе

Овај рад покушава да премости неколико празнина за које мислимо да тренутно постоје у формализацији геометрије.

Због своје важности, геометрија комплексних бројева је добро описана у литератури. Постоје многи уџбеници који описују ову област са много детаља (током нашег рада, ми смо интензивно користили уџбенике које су писали Нидам (енг. *Needham*) [130] и Швердфегер (нем. *Schwerdtfeger*) [148]). Такође, постоји велики избор материјала за ову област (слајдова, белешки,

приручника) који су доступни на вебу. Ипак, ми нисмо упознати да постоји формализација ове области и у овом раду, ми представљамо наше потпуно формално, механички проверено представљање геометрије комплексне равни које је, према нашем сазнању, прво такве врсте.

Додатно, ми сматрамо да је једнако (или чак више) важно искуство које смо стекли приликом различитих покушаја да достигнемо коначни циљ од коначног резултата. Наиме, постоји много различитих начина на које је област изложена у литератури. На пример, Нидам [130] и Швердфегер [148], представљају два врло различита начина приказивања исте приче — један приступ је више геометријски оријентисан, док је други више алгебарски оријентисан. Наше искуство показује да је избор правог приступа важан корак у остваривању циља да формализација буде спроводљива у оквиру асистента за доказивање теорема — и показало се да што је више приступ алгебарски оријентисан, то је формализација једноставнија, лепша, флексибилнија и робуснија.

У оквиру рада на докторској тези, формализована је аналитичка геометрија Декартове равни, геометрија комплексне равни, дат је део формализације Поенкареовог диск модела, дата је формална анализа алгебарских метода и систем за аутоматско доказивање у стереометрији. У наставку текста набројани су основи доприноси тезе:

- Формализована је аналитичка геометрија тј. Декартова раван у оквиру система за интерактивно доказивање теорема. Представљена је добро изграђена формализација Декартове геометрије равни у оквиру система *Isabelle/HOL*. Дато је неколико различитих дефиниција Декартове координатне равни и доказано је да су све дефиниције еквивалентне. Дефиниције су преузете из стандардних уџбеника, али је подигнут ниво ригорозности. На пример, у текстовима се обично не помињу појмови као што су релација еквиваленције и класа еквиваленције које ће морати да буду уведене у формалним дефиницијама. Формално је доказано да Декартова координатна раван задовољава све аксиоме Тарског и већину аксиома Хилберта (укључујући и аксиому непрекидности). Анализирани су докази и који од два система аксиома је лакши за формализацију. Формално је доказано да је аналитичка геометрија модел синтетичке геометрије и анализирано је колико су докази заиста једноставни.

- Коначни резултат нашег рада је добро развијена теорија проширене комплексне равни (дате као комплексна пројективна права, али и као Риманова сфера), њених објеката (кругови и праве) и њених трансформација (на пример, инверзија или Мебијусових трансформација). Ова формализација може да служи као веома важан блок за изградњу будућих формалних модела различитих геометрија (нпр, наша мотивација за овај рад је била управо у покушају да се формализује Поенкареов диск модел хиперболичке геометрије). Већина концепата које смо формализовали већ је описана у литератури (иако је постојало много детаља које смо морали да попунимо јер их нисмо нашли у литератури који смо разматрали). Ипак, наш рад је захтевао обједињавање различитих извора у једну јединствену, формалну репрезентацију и пребацивање у један јединствен језик узевши да су описи били првобитно дати на много различитих начина. На пример, чак и у оквиру истог уџбеника, без икаквог формалног оправдања, аутори често лако прелазе из једне поставке у другу (рецимо, из обичне комплексне равни у проширену комплексну раван), прелазе између геометријског и алгебарског представљања, често користе многе недоказане, нетривијалне чињенице (посматрајући их као део математичког "фолклора") и др. Један од наших најзначајнијих доприноса је управо расветљавање ових непрецизности и креирање униформног, јасног и самосталног материјала.

- Извршена је формализација шест аксиома Тарског у оквиру Поенкареовог диск модела. Дата је дефиниција релације *између* и доказана су нека њена основна својства у оквиру Поенкареовог диск модела.

- У циљу изградње формално верификованог система за аутоматско доказивање у геометрији који користи метода Гребнерових база или Вуову методу, имплементиран је корак превођења планиметријских тврђења у алгебарску форму у оквиру система *Isabelle/HOL*. Поред тога, направљен је и прототип система за доказивање тврђења у стереометрији (на чијој се верификацији и даље ради). Први корак је представити стереометријске објекте и тврђења у одговарајућем облику коришћењем полинома. Направљен је софтвер који омогућава опис стереометријских конструкција и тврђења у једноставном облику који је разумљив човеку, а потом превођење тог записа у систем полинома на који се примењује

Вуова метода или метода Гребнерових база. Систем је тестиран кроз неколико различитих задатака из уџбеника за средње школе и факултете и задатака са математичких такмичења, и анализирана је ефикасност оваквог приступа.

1.3 Организација тезе

Остатак тезе организован је на следећи начин. Глава 2 садржи преглед развоја интерактивних доказивача теорема. Дате су теоријске основе и описани основни принципи технологије интерактивних доказивача теорема. Описане су главне карактеристике система *Isabelle/HOL* и дати су бројни примери. У глави 3 наведени су неки најзначајнији радови у формализацији геометрије. Такође, представљени су и радови из аутоматског доказивања у геометрији. У глави 4 дати су резултати формализације аналитичке геометрије и доказано је да она представља модел аксиома Тарског (поглавље 4.3) и модел аксиома Хилберта (поглавље 4.4). Глава 5 приказује формализацију проширене комплексне равни, описане су трансформације проширене комплексне равни и доказано је да неке аксиоме Тарског важе у Поенкареовом диск моделу. У Глави 6 представљени су алгебарски методи за аутоматско доказивање у геометрији, дата је формализација алгебризације геометријских тврђења за планарну геометрију и предложена је алгебризација геометријских тврђења у стереометрији. Предложени систем је тестиран над више примера и представљени су резултати. Најзад, у глави 7 сумирани су закључци и наведени неки правци будућег рада.

Глава 2

Интерактивни доказивачи теорема

У овом поглављу ћемо приказати технологију интерактивних доказивача теорема.

2.1 Неколико речи о λ–рачуну и Кари–Хауард изоморфизму

Кари–Хауард изоморфизам даје везу између система формалне логике која се може наћи у теорији доказа и рачуна теорије типова. На пример, исказна логика одговара једноставном λ–рачуну са типовима, логика првог реда одговара зависним типовима (енг. *dependent types*), логика другог реда одговара полиморфним типовима итд. Изоморфизам има бројне аспекте: формуле одговарају типовима, докази одговарају термовима, нормализација доказа одговара редукцији термова.

Крајем 1920. године у Черчовом раду „Рачун ламбда–конверзије" (енг. *The calculi of lambda-conversion*) [37] представљен је λ–рачун, систем чији је циљ био да опише особине функционалне апстракције, примене и супституције, односно да преброди ограничења која су имали Раселова теорија типова и Цермел–Френкелова теорија скупова. Систем је био без типова, без закона о изузимању трећег, али са неограниченом квантификацијом и експлицитним формалним правилима λ–конверзије. Ипак, убрзо након објављивања показало се да је систем контрадикторан, односно да није логички конзистентан. Да

би се елиминисала неконзистентност, систем је проширен типовима [36]. Показана је конфлуентност λ–система и показано је да нумеричке λ–дефинисане функције одговарају Тјуринговим израчунљивим функцијама, што је са другим сродним резултатима сугерисало чувену Черчову тезу која тврди да *Класа интуитивно израчунљивих функција идентична је са класом формално израчунљивих функција.* Током 1930–их λ–рачун је послужио за описивање израчунавања (алгоритама) и за доказ неодлучивости логике првог реда [36]. λ–рачун са типовима има мању моћ израчунавања од λ–рачуна, али је логички конзистентан, па је послужио као основа неких логика и неких програмских језика. Постоји много формализама λ–рачуна са типовима и ми ћемо овде укратко представити само неке најзначајније за ову тезу.

У овом поглављу представићемо теоријске основе Кари–Хауард изоморфизма, неке основне особине и последице. Више информација се може пронаћи у [10, 153].

λ–рачун

Черч и Кари су λ–рачун и системе комбинаторне логике први пут предложили 1930. године. Систем који се састоји од λ–термова и β–редукције се показао веома корисним за формализацију интуитивног појма ефективне израчунљивости. Сви ови резултати су били инспирација за развијање *теорије рекурзивних функција.* λ–рачун се показао као важна алатка за дизајн, имплементацију и теорију више програмских језика.

λ–термови

Дефиниција 2.1.1. Нека је $V = \{v_0, v_1, \ldots\}$ скуп променљивих. Скуп Λ *термова* је скуп ниски који се дефинише на следећи начин:

$$\Lambda = V \mid (\Lambda\Lambda) \mid (\lambda V.\ \Lambda)$$

(i) $x, y, z, \ldots$ (елементи V) означавају променљиве

(ii) $M, N, L, \ldots$ означавају λ–термове

(iii) Терм $\lambda x.\ M$ означава *апстракцију* (над променљивом x)

(iv) Терм облика $(M\ N)$ означава *примену* (M над N)

***Пример* 2.1.1.** *Пример λ–термова:*

(i) $\lambda x.x\ y$

(ii) $(\lambda x.x\ x)\ \lambda y.y\ y$

Дефиниција 2.1.2. За λ–терм $M \in \Lambda$ дефинишемо скуп $FV(M) \subseteq V$ слободних променљивих λ–терма M на следећи начин:

$$\begin{aligned} FV(x) &= \{x\}; \\ FV(\lambda x.P) &= FV(P)\backslash\{x\}; \\ FV(P\ Q) &= FV(P) \cup FV(Q). \end{aligned}$$

Ако је $FV(M) = \{\}$, онда је M *затворен.*

Дефиниција 2.1.3. За λ–термове $M, N \in \Lambda$ и променљиву $x \in V$ *замена* променљиве x са N у M, означено са $M[x := N]$ се дефинише на следећи начин:

$$\begin{aligned} x[x := N] &= N; \\ y[x := N] &= y \ \text{ ако је } \ x \neq y; \\ (P\ Q)[x := N] &= P[x := N]Q[x := N]; \\ (\lambda y.P)[x := N] &= \lambda y.P[x := N] \ \text{ ако важи } \ x \neq y \text{ и } \ y \notin FV(N). \end{aligned}$$

Редукција

Дефиниција 2.1.4. Нека је $\rightarrow_\beta$ најмања релација над Λ таква да

$$(\lambda x.P)\ Q \rightarrow_\beta P[x := Q]$$

која задовољава следећа правила:

$$\begin{aligned} &P \rightarrow_\beta P' \Rightarrow \forall x \in V : \lambda x.P \rightarrow_\beta \lambda x.P'; \\ &P \rightarrow_\beta P' \Rightarrow \forall Z \in \Lambda : P\ Z \rightarrow_\beta P'\ Z; \\ &P \rightarrow_\beta P' \Rightarrow \forall Z \in \Lambda : Z\ P \rightarrow_\beta Z\ P'. \end{aligned}$$

Дефиниција 2.1.5. Релација $\twoheadrightarrow_\beta$ (β–редукција у више корака) је транзитивно и рефлексивно затворење релације $\rightarrow_\beta$, односно, $\twoheadrightarrow_\beta$ је најмања релација која задовољава следећа правила:

$$P \twoheadrightarrow_\beta P' \twoheadrightarrow_\beta P'' \wedge P' \Rightarrow P \twoheadrightarrow_\beta P''$$
$$P \to_\beta P' \Rightarrow P \twoheadrightarrow_\beta P'$$
$$P \twoheadrightarrow_\beta P.$$

***Пример* 2.1.2.** *(i)* $(\lambda x.x\ x)\lambda z.z \to_\beta (x\ x)[x := \lambda z.z] = (\lambda z.z)\lambda y.y$

(ii) $(\lambda z.z)\lambda y.y \to_\beta z[z := \lambda y.y] = \lambda y.y$

(iii) $\lambda x.x\ x)\lambda z.z \twoheadrightarrow_\beta \lambda y.y$

Дефиниција 2.1.6 (конфлуентност)**.** Релација $\to$ је конфлуентна ако за свака три λ–терма $M_1, M_2, M_3 \in \Lambda$, ако $M_1 \to M_2$ и $M_1 \to M_3$ онда постоји $M_4 \in \Lambda$ такав да $M_2 \to M_4$ и $M_3 \to M_4$.

***Теорема* 2.1.1** (конфлуентност)**.** [1] *Релација* $\twoheadrightarrow_\beta$ *је конфлуентна.*

Често се λ–терм M_4 назива *нормалном формом* λ–терма M_1, а поступак налажења нормалне форме се назива *нормализација.*

Дефиниција 2.1.7. Релација $\to_\eta$ је најмања релација која задовољава следећа својства:

1. Ако $x \notin FV(M)$ онда $\lambda x.Mx \to_\eta M$
2. Ако $P \to_\eta P'$ онда $\lambda x.P \to_\eta \lambda x.P'$
3. Ако $P \to_\eta P'$ онда $PQ \to_\eta P'Q$ и $QP \to_\eta QP'$

Релација η–редукција је такође конфлуентна.

Релација β–редукција представља примену функције, а η–редукција служи да омогући проверу да су две функције једнаке ако и само ако имају једнаке вредности за све аргументе.

Интуиционистичка логика и природна дедукција

Природна дедукција је формални дедуктивни систем који је развио Гентцен 1930. године [54, 55] чији циљ је био да формални докази буду слични резоновању у традиционалним математичким текстовима.

[1] *Доказ у [10].*

Постоји систем природне дедукције за класичну логику и систем природне дедукције за интуиционистичку логику. Систем природне дедукције за класичну логику има једну аксиоматску схему, $A \vee \neg A$ (искључење трећег), док систем за интуиционисичку логику нема аксиома.

За сваки логички везник постоје правила која га уводе (правила I–типа) и правила која га елиминишу (правила E–типа). Додатно, постоји правило (правило efq, *ex falso quodlibit*) које не елиминише нити уводи неки логички везник. Током извођења доказа у систему природне дедукције могу се користити недоказане претпоставке, али оне морају бити елиминисане пре краја извођења. Претпоставка се записује коришћењем $[\_]$. У табели 2.1 су дата правила извођења система природне дедукције и за класичну и за интуиционистичку логику.

$$\frac{\begin{array}{c}[A]^u \\ \vdots \\ \bot\end{array}}{\neg A}\,\neg I, u \qquad \frac{A \quad \neg A}{\bot}\,\neg E$$

$$\frac{A \quad B}{A \wedge B}\,\wedge I \qquad \frac{A \wedge B}{A}\,\wedge E \qquad \frac{A \wedge B}{B}\,\wedge E$$

$$\frac{A}{A \vee B}\,\vee I \qquad \frac{B}{A \vee B}\,\vee I \qquad \frac{A \vee B \qquad \begin{array}{c}[A]^u \\ \vdots \\ C\end{array} \qquad \begin{array}{c}[B]^v \\ \vdots \\ C\end{array}}{\mathrm{C}}\,\vee E, u, v$$

$$\frac{\begin{array}{c}[A]^u \\ \vdots \\ B\end{array}}{A \Rightarrow B}\,\Rightarrow I, u \qquad \frac{A \quad A \Rightarrow B}{B}\,\Rightarrow E$$

$$\frac{\bot}{D}\,e\ f\ q$$

Табела 2.1: Правила извођења система природне дедукције за класичну или интуиционистичку исказну логику

У систему природне дедукције доказ је стабло чијем је сваком чвору придружена формула. Формула је *теорема* природне дедукције ако постоји доказ у чијем је корену A и који нема неослобођених претпоставки и тада пишемо $\vdash A$ и кажемо да је формула A доказива у систему природне дедук-

ције. Ако постоји доказ у чијем корену је формула A и који има неослобођене претпоставке које припадају неком скупу Γ, онда кажемо да је формула A *дедуктивна последица* скупа претпоставки Γ и тада пишемо $\Gamma \vdash A$.

Постоји много различитих ознака коришћених да прикажу правила и доказе природне дедукције. Ми ћемо овде представити нека правила природне дедукције за интуиционистичку исказну логику. Синтакса интуиционистичке исказне логике је слична као и синтакса класичне исказне логике. Скуп формула Φ можемо дефинисати индуктивно, коришћењем V, бесконачног скупа исказних променљивих:

$$\Phi = \bot \mid V \mid (\Phi \to \Phi) \mid (\Phi \wedge \Phi) \mid (\Phi \vee \Phi)$$

Интуиционистичка логика [97, 135] је ослабљена у односу на класичну логику пре свега изузимајући принцип о изузимању трећег. У светлу Кари–Хауард изоморфизма хипотезе A, B, C, ..., се сматрају проблемима које треба решити, а њихови докази a, b, c, ..., методама који их решавају. Конструктивни доказ се може посматрати на следећи начин.

i) Доказ за $A \wedge B$ је пар (a, b) где је a доказ за A, а b доказ за B.

ii) Доказ за $A \to B$ је функција f која за сваки доказ a за A даје доказ $f(a)$ за B.

iii) Доказ за $A \vee B$ је доказ a за или доказ b за B.

iv) Не постоји доказ за $\bot$.

Уместо формула, чворови стабла извођења су секвенти облика $\Gamma \vdash F$ при чему је Γ коначан скуп претпоставки. У табели 2.2 се могу видети правила извођења система природне дедукције за део интуиционистичке исказне логике која су задата у таквом облику.

Једноставан λ–рачун са типовима

λ–термови немају фиксни домен, али су Кари [43] и Черч [36] увели и верзију система са типовима, независно један од другог. Зато постоје две верзије, λ–рачун са типовима 'а ла Кари и λ–рачун са типовима 'а ла Черч. Иако су ови системи различити, суштински представљају исте идеје и постоје исти закључци. Ипак, треба бити пажљив, јер нису у потпуности исти, али

$$\frac{\alpha \in \Gamma}{\Gamma \vdash \alpha}\ Ax \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\ \bot_E \qquad \frac{}{\Gamma \vdash \top}\ \top_I$$

$$\frac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta}\ \to_I \qquad \frac{\Gamma \vdash \alpha \to \beta \qquad \Gamma \vdash \alpha}{\Gamma \vdash \beta}\ \to_E$$

$$\frac{\Gamma \vdash \alpha \qquad \Gamma \vdash \beta}{\Gamma \vdash \alpha \wedge \beta}\ \wedge_I \qquad \frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \alpha}\ \wedge_{E_1} \qquad \frac{\Gamma \vdash \alpha \wedge \beta}{\Gamma \vdash \beta}\ \wedge_{E_2}$$

$$\frac{\Gamma \vdash \alpha}{\Gamma \vdash \alpha \vee \beta}\ \vee_{I_1} \qquad \frac{\Gamma \vdash \beta}{\Gamma \vdash \alpha \vee \beta}\ \vee_{I_2} \qquad \frac{\Gamma \vdash \alpha \vee \beta \qquad \Gamma, \alpha \vdash \gamma \qquad \Gamma, \beta \vdash \gamma}{\Gamma \vdash \gamma}\ \vee_E$$

$$\frac{\Gamma \vdash \alpha \qquad \Gamma \vdash \neg\alpha}{\Gamma \vdash \bot}\ \neg_E \qquad \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \neg_I$$

Табела 2.2: Правила извођења система природне дедукције за интуиционистичку исказну логику

анализу тих разлика (због обима) нећемо дати у овој тези. У наставку ће λ–рачун са типовима бити представљен 'а ла Кари.

Сваком терму се придружује неки тип. Тип терма зависи од типа променљивих које се у њему јављају. Ако су типови променљивих одређени контекстом Γ тада је могуће одредити и тип терма. Правила одређивања типова се означавју релацијом $\vdash$. Ако $\Gamma \vdash M : \sigma$ онда кажемо да *λ–терм M има тип σ у* Γ.

Дефиниција 2.1.8.

(i) Нека скуп U означава пребројиво бесконачан алфабет чији чланови се зову *типске променљиве*. Скуп *једноставних типова* Π је скуп ниски који су дефинисани граматиком:

$$\Pi = U \mid (\Pi \to \Pi)$$

Обично користимо симболе $\alpha, \beta, \ldots$ да означимо произвољне типске променљиве, а $\tau, \sigma, \ldots$ да означимо произвољне типове. Неформално, $\sigma \to \tau$ означава скуп функција које сликају σ у τ. На пример, $\vdash \lambda x.x : \sigma \to \sigma$ неформално изражава да идентитет је функција која слика одређени скуп у себе самог.

(ii) *Контекст* C одређује типове променљивих и дефинише се као скуп парова облика

$$\{x_1 : \tau_1, \ldots, x_n : \tau_n\}$$

при чему $\tau_1, \ldots, \tau_n \in \Pi$, $x_1, \ldots, x_n \in V$ (променљиве за Λ) и важи $x_i \neq x_j$ за свако $i \neq j$.

(iii) *Домен* Γ за контекст $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$ се дефинише са:

$$\mathrm{dom}(\Gamma) = \{x_1, \ldots, x_n\}$$

Унију $\Gamma \cup \Gamma'$ означавамо Γ, Γ' ако важи $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \{\}$.

(iv) *Кодомен* контекста $\Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\}$ се дефинише са:

$$|\Gamma| = \{\tau \in \Pi \mid (x : \tau) \in \Gamma, \text{за неко } x\}$$

(v) *Релација* $\vdash$ над $C \times \Lambda \times \Pi$ је дефинисана типским правилима:

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash (\lambda x : \tau.M)(\tau \to \sigma)} \qquad \frac{\Gamma \vdash M : \tau \to \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M\ N : \sigma}$$

при томе за прво и друго правило важи $x \notin \mathrm{dom}(\Gamma)$.

(vi) Једноставан λ–рачун са типовима $\lambda^{\to}$ је тројка $(\Lambda, \Pi, \vdash)$.

Једноставан λ-рачун је основа за програмске језике као што су *Lisp* и *Scheme*, а различите варијанте једноставног λ–рачуна са типовима су послужиле као основа за програмске језике *Pascal*, *ML* и *Haskell*.

***Теорема* 2.1.2** (Редукција субјекта)**.** [2] *Ако* $\Gamma \vdash M : \sigma$ *и* $M \to_\beta N$, *онда* $\Gamma \vdash N : \sigma$.

И у λ–рачуну са типовима β–редукција ($\to_\beta$) је конфлуентна.

***Теорема* 2.1.3** (Конфлуентност)**.** [3] *Претпоставимо да важи* $\Gamma \vdash M : \sigma$. *Ако* $M \twoheadrightarrow_\beta N$ *и* $M \twoheadrightarrow_\beta N'$, *онда постоји* L *такав да* $N \twoheadrightarrow_\beta L$ *и* $N' \twoheadrightarrow_\beta L$ *и* $\Gamma \vdash L : \sigma$.

***Теорема* 2.1.4** (Строга нормализација)**.** [4] *Ако важи* $\vdash M : \sigma$, *онда бесконачана редукција* $M_1 \to_\beta M_2 \to_\beta \ldots$ *не постоји.*

Редукција субјекта, конфлуенција и строга нормализација обезбеђују да се било која редукција типског λ–терма завршава у нормалној форми истог типа, при чему је нормална форма независна од редоследа примене правила. Ова својства су веома важна јер ако би λ–рачун са типовима посматрали

[2] *Доказ у [10].*
[3] *Доказ у [10].*
[4] *Доказ у [10].*

као идеализовани програмски језик, онда би β–редукција била један корак у израчунавању. Свакако бисмо желели да се извршавање након неког броја корака заврши, а не да се одвија бесконачно, а то нам обезбеђује строга нормализација. Са друге стране, системи који задовољавају строгу нормализацију нису Тјуринг–комплетни тј. у таквим системима није могуће записати све израчунљиве функције.

Кари–Хауард изоморфизам

Кари–Хауард изоморфизам представља запањујућу и важну аналогију између λ–рачуна и правила природне дедукције за исказну интуиционистичку логику. Било који доказ у импликацијском фрагменту исказне интуиционистичке логике (интуиционистичка логика само са везником имликација, означено са IPC($\rightarrow$)) одговара неком типском λ–терму и обратно.

Ако је V (скуп предикатских променљивих) једнак U (скуп типских променљивих), онда су Φ (скуп исказних формула имлицитног фрагмента исказне интуиционистичке логике) и Π (скуп једноставних типова) једнаки.

***Теорема* 2.1.5** (Кари–Хауард изоморфизам)**.** [5]

(i) Ако $\Gamma \vdash M : \varphi$ *онда* $|\Gamma| \vdash \varphi$.

(ii) Ако $\Gamma \vdash \varphi$ *онда постоји* $M \in \Lambda_{\Pi}$ *такав да* $\Delta \vdash M : \varphi$, *при чему је* $\Delta = \{(x_\varphi : \varphi) \mid \varphi \in \Gamma\}$.

Коришћење правила Ax интуиционистичке исказне логике одговара променљивима у термовима, коришћење $\rightarrow_E$ правила одговара примени, а $\rightarrow_I$ правила одговара апстракцији, термови служе као линеарна репрезентација стабла доказа. Посматрајмо упоредо правила дата на табели 2.3:

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{\alpha \in \Gamma}{\Gamma \vdash \alpha}\, Ax$$

$$\frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash (\lambda x : \tau . M)(\tau \rightarrow \sigma)} \qquad \frac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \rightarrow \beta} \rightarrow_I$$

$$\frac{\Gamma \vdash M : \tau \rightarrow \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash M\ N : \sigma} \qquad \frac{\Gamma \vdash \alpha \rightarrow \beta \qquad \Gamma \vdash \alpha}{\Gamma \vdash \beta} \rightarrow_E$$

Табела 2.3: Правила природне дедукције наспрам правила за λ–рачун

[5] *Доказ у [10].*

Табела која приказује однос λ–рачуна са типовима и имплицитног фрагмента исказне интуиционистичке логике:

$\lambda^{\to}$	**IPC**$(\to)$
терм променљива	претпоставка
терм	доказ (конструкција)
типска променљива	исказна променљива
тип	формула
конструктор типа	везник
да ли постоји терм датог типа?	да ли постоји доказ (конструкција) дате претпоставке?
типски терм	конструкција дате претпоставке
редукција ($\beta\eta$–редукција)	нормализација

Провера типа

Дефиниција 2.1.9.

(i) Проблем *провере типа* (енг. *type checking*) је проблем одлучивања да ли важи $\Gamma \vdash M : \tau$ за дати контекст Γ, терм M и тип τ.

(ii) Проблем *реконструкције типа* (енг. *type reconstruction*) је проблем одлучивања да ли за дати терм M постоји контекст Γ и тип τ такав да важи $\Gamma \vdash M : \tau$.

(iii) Проблем *празног типа* (енг. *type emptiness*) или *настањености типа* (енг. *type inhabitation*) је проблем одлучивања да ли за дати тип τ постоји затворен терм M такав да важи $\vdash M : \tau$.

На основу Кари–Хауард изоморфизма, σ се може схватити као логичка формула, а M као доказ. Зато проблем провере типа одговара проблему провере доказа, тј. интерактивном доказивању теорема, а проблем попуњености типа одговара провери да ли постоји барем један доказ, тј. аутоматском доказивању теорема.

Одлучивост ових проблема зависи од коришћене верзије λ–рачуна. Једноставан λ–рачун са типовима према Кари–Хауард изоморфизму одговара интуиционистичкој исказној логици.

***Теорема* 2.1.6.** [6] *Проблем празног типа за једноставан λ–рачун са типовима је еквивалентан проблему испитивања ваљаности у имплицитном фрагменту интуиционистичке исказне логике.*

Зато се за једноставни λ–рачун са типовима могу доказати следећа тврђења.

***Теорема* 2.1.7.** [7] *Проблем провере типа у једноставном λ–рачуну са типовима је одлучив.*

***Теорема* 2.1.8.** [8] *Проблем празног типа у једноставном λ–рачуну са типовима је одлучив и P–комплетан.*

***Теорема* 2.1.9.** [9] *Проблем реконструкције типа за једноставан типски λ–рачун је P–комплетан.*

Ова својства је интересантно повезати са Кари–Хауард изоморфизмом. Наиме, ако је неки исказ доказив у неком систему, онда ће у λ–рачуну одговарајући тип бити попуњен, а терм који настањује (енг. *inhabiting*) тај тип је заправо посматран као његов доказ. Зато нам је важна одлучивост проблема празног типа. Оно што је још интересантније, из угла програмског језика, терм који настањује тип се може посматрати као извршив програм.

Зависни типови и полиморфни λ–рачун

Посматрано са програмерске тачке, зависни типови (енг. *dependent types*) су они типови који *зависе* од вредности објекта. На пример, посматрајмо тип `string(n)` који представља све бинарне ниске дужине `n` [10]. Овај тип зависи од избора `n:int`. Оператор `string` прави тип над целим бројевима и одговара, преко Кари-Хауард изоморфизма, предикату над типом `int`. Такав предикат се зове *конструктор типа* или само *конструктор*. Потребно је класификовати конструкторе према њиховом домену и то доводи то појма *род* (енг. kind): кажемо да је конструктор `string` рода $int \Rightarrow *$, при чему је $*$ род свих типова. Наравно, предикати не морају бити само бинарни, те род може

[6] *Доказ у [10].*
[7] *Доказ у [10].*
[8] *Доказ у [10].*
[9] *Доказ у [10].*
[10] Пример из [124].

укључивати и $\tau_1 \Rightarrow \ldots \Rightarrow \tau_n \Rightarrow *$. Претпоставимо, на пример, да `string(n)` враћа ниске дужине `n` који се састоје само из нула. Тип ове процедуре би био $(\forall n : \texttt{int})\texttt{string}(n)$.

Уопштено, тип облика $(\forall x : \tau)\sigma$ је тип функције која се примењује на објекте типа τ, а враћа објекте типа $\sigma[x := a]$ за сваки аргумент $a : \tau$. Ово понашање је заправо исто као и понашање $\tau \to \sigma$.

Дефиниција 2.1.10. Ако $\Gamma \vdash \tau : *$, онда кажемо да је τ *тип* у контексту Γ.

У наредним поглављима ћемо представити три система који су проширења једноставног λ–рачуна са типовима, а потом ћемо дати везу између различитих система коришћењем λ–коцке. Ова проширења омогућавају увођење комплекснијих типова, односно ширу и специфичнију репрезентацију података, али истовремено нарушавају одлучивост проблема провере типа и настањености типа.

Систем $\lambda 2$

Систем $\lambda 2$ је проширење λ–рачуна за логику другог реда. Овим системом уводи се полиморфизам, тј. овим системом су омогућене две зависности, термови да зависе од термова (ово важи и у λ–рачуну) и термови да зависе од типова.

Дефиниција 2.1.11.

(i) *Тип* (другог реда) се дефинише:

- Типске промењљиве су типови
- Ако су σ и τ типови, онда је и $\sigma \to \tau$ тип
- Ako је σ тип, а α типска промењљива, онда $\forall\alpha\sigma$ је тип

(ii) Добро типизирани λ–термови су дефинисани правилима за извођење типа који су дати у табели 2.4. Сваки терм је или промењљива, обична примена или апстракција или је

- *полиморфна апстракција*, записана као $\Lambda\alpha.M$, при чему је M терм, а α типска промењљива или
- *примена типа*, записана као $(M\tau)$, при чему је M терм, а τ је тип.

$\Gamma, x : \tau \vdash x : \tau$

$$\frac{\Gamma N : \tau \to \sigma \qquad \Gamma \vdash M : \tau}{\Gamma \vdash NM : \sigma} \qquad \frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash \lambda x.M : \tau \to \sigma}$$

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash (\Lambda \alpha M) : \forall \alpha \sigma} \alpha \notin FV(\Gamma) \qquad \frac{\Gamma \vdash M : \forall \alpha \sigma}{\Gamma \vdash M\tau : \sigma[\alpha := \tau]}$$

Табела 2.4: Правила за извођење типа

Неформално, за полиморфну апстракцију $\Lambda\alpha.M$, терм M можемо посматрати као полиморфна процедура са параметром типа α.

Пример да терм зависи од терма може бити следећи $\lambda m : A.F : A \to B$. У програмском језику C++ ова зависност се често може приметити, рецимо у аритметичким изразима `a = b + 1`, а где терм `a` зависи од терма `b + 1`.

Посматрајмо функцију идентитета, тј. функцију која за дати улаз враћа непромењени излаз. Над природним бројевима, `nat` тип функције је $\lambda x : nat.x$, над буловским типом, тип функције је $\lambda x : bool.x$, а над типом $nat \to bool$ тип функције је $\lambda x : nat \to bool.x$. Значи, у зависности од типа можемо имати много функција идентитета, али питање је како дефинисати општу функцију индентитета, односно, ако имамо произвољан тип α функција би била $f \equiv \lambda x : \alpha.x$. Наравно, сада не можемо да пишемо fM јер овај терм није легалан, али можемо да мало изменимо дефиницију, односно да у систему $\lambda 2$ дамо дефиницију: $\lambda\alpha : k.\lambda x : \alpha.x$, односно, α је неки тип из скупа свих типова рода k. Сада имамо терм који зависи од типа. Пример у језику C++ би могла да буде функција која је дефинисана над неким `template` типом.

Систем $\lambda 2$ преко Кари–Хауарад изоморфизма одговара исказној логици другог реда [58, 59, 60]. Коришћењем овог својства може се доказати да важи:

***Теорема* 2.1.10.** [11] *Провера типа, проблем настањености типа и проблем реконструкције типа су неодлучиви у систему* $\lambda 2$.

Систем $\lambda\underline{\omega}$

Систем $\lambda\underline{\omega}$ је проширење у коме се може направити зависност „тип зависи од типа", на пример $f \equiv \lambda\alpha : *.\alpha \to \alpha$. Пример у програмирању би био тип `vector<int>` који зависи од типа `vector` који је заправо дефинисани `template`.

[11] *Докази у [10].*

Питање је шта је f, јер није терм, није ни тип. Зато се уводи $\mathcal{K} = * \mid \mathcal{K} \to \mathcal{K}$, тј. $\mathcal{K} = \{*, * \to *, * \to * \to *, \ldots\}$. Уводи се и ознака $\Box$, при чему $k : \Box$ означава да је k добро формирани род, односно да $k \in \mathcal{K}$, а при томе само $\Box$ није део језика. Видимо да $\vdash (\lambda\alpha : *.\alpha \to \alpha) : (* \to *)$, односно f је конструктор рода $* \to *$.

Дефиниција 2.1.12. Нека је V скуп променљивих, а $C = \{*, \Box\}$ скуп константи. Типови и термови $\mathcal{T}$ у систему $\lambda\underline{\omega}$ дефинишу се на следећи начин:

$$T = V \mid C \mid \mathcal{T}\mathcal{T} \mid \lambda V : \mathcal{T}.\mathcal{T} \mid \mathcal{T} \to \mathcal{T}$$

Релација $\vdash$ (тј. $\vdash_{\lambda\underline{\omega}}$, $\Gamma \vdash_{\lambda\underline{\omega}} M \ : \ A$) се дефинише правилима датим у табели 2.5, при чему променљива s може узети било коју вредност из скупа $\{*, \Box\}$.

$\vdash * : \Box$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}\ x \notin dom(\Gamma) \qquad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}\ x \notin dom(\Gamma)$$

$$\frac{\Gamma \vdash A : s \qquad \Gamma \vdash B : s}{\Gamma \vdash (A \to B) : s} \qquad \frac{\Gamma \vdash F : (A \to B) \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B}$$

$$\frac{\Gamma, x : A \vdash b : B \qquad \Gamma \vdash (A \to B) : s}{\Gamma \vdash (\lambda x : A.b) : (A \to B)} \qquad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B \twoheadrightarrow_\beta B'}{\Gamma \vdash A : B'}$$

Табела 2.5: Правила извођења

Систем λP

Систем λP зависних типова је проширење једноставног λ–рачуна са типовима. Са овим системом могуће је креирати зависност „тип зависи од терма". Пример би био $A^{m \times n}$, тип свих матрица димензије $m \times n$, при чему овај тип зависи од бројева m и n.

Дефиниција 2.1.13. Нека је V скуп променљивих, $C = \{*, \Box\}$ скуп константи, а типови и термови $\mathcal{T}$ у систему $\lambda\underline{\omega}$ дефинишу се на следећи начин:

$$T = V \mid C \mid \mathcal{T}\mathcal{T} \mid \lambda V : \mathcal{T}.\mathcal{T} \mid \Pi V : \mathcal{T}.\mathcal{T}$$

При томе, Π представља декартовски производ. Релација $\vdash$ (тј. $\vdash_{\lambda P}$) се дефинише правилима из табеле 2.6, при чему променљива s може бити било која вредност из скупа $\{*, \Box\}$.

$$\vdash * : \Box$$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}\, x \notin dom(\Gamma) \qquad \frac{\Gamma \vdash A : B \qquad \Gamma C : s}{\Gamma, x : C \vdash A : B}\, x \notin dom(\Gamma)$$

$$\frac{\Gamma \vdash A : * \qquad \Gamma, x : A \vdash B : s}{\Gamma \vdash (\Pi : A.B) : s} \qquad \frac{\Gamma \vdash F : (\Pi x : A.B) \qquad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[x := a]}$$

$$\frac{\Gamma, x : A \vdash b : B \qquad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.b) : (\Pi x : A.B)} \qquad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s \qquad B \twoheadrightarrow_\beta B'}{\Gamma \vdash A : B'}$$

Табела 2.6: Правила извођења

Систем λP преко Кари–Хауард изоморфизма одговара предикатској логици првог реда.

***Теорема* 2.1.11.** [12] *Проблем настањености типа у λP је неодлучив. Проблем реконструкције типа је одлучив. Проблем провере типа је неодлучив.*

λ–коцка

Већ смо видели да је у различитим системима могуће направити различите зависности.

У систему $\lambda^{\rightarrow}$ можемо имати терм

$$\vdash \lambda x : \sigma.x : \sigma \to \sigma.$$

За дати терм M можемо формирати нови терм $\lambda x : \sigma.M$ који очекује терм као аргумент, другим речима овај *терм зависи од терма*.

У систему $\lambda 2$ можемо имати терм

$$\vdash \Lambda\alpha : *.\lambda x\alpha.x : \forall\alpha.\alpha \to \alpha.$$

За дати терм M, можемо формирати нови терм $\Lambda\alpha : *.M$ који очекује тип σ као аргумент, другим речима *терм зависи од типа*.

Коначно, у систему λP можемо имати израз

$$\alpha : * \vdash \lambda x : \alpha.\alpha : \alpha \Rightarrow *$$

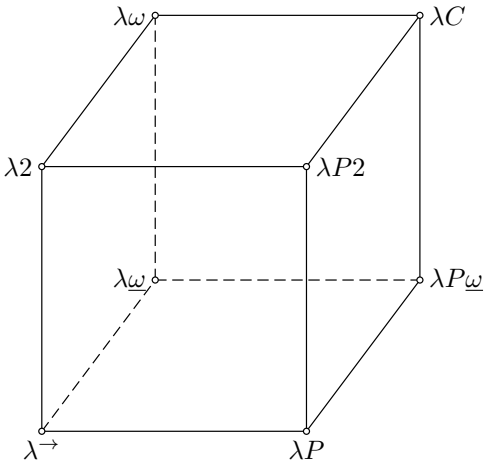
што изражава да конструктор очекује терм типа α, а конструише члан $*$. Ако применимо овај израз на терм α, онда добијамо тип $\alpha : *, y : \alpha \vdash (\lambda x :$

[12] *Докази у [10].*

$\alpha.\alpha)y : *$. Односно, за дати тип α можемо формирати конструктор $\lambda x : \alpha.\alpha$ који очеекује терм типа α као аргумент, другим речима *конструктор типа зависи од терма*.

Можемо направити и да *тип зависи од типа*, израз облика $\lambda\alpha : *.\alpha \to \alpha$. Систем у коме је то могуће је $\lambda\underline{\omega}$.

Слично, постоје и други системи који имају исте или друге зависности и све их повезује λ–коцка, видети слику 2.1.



Слика 2.1: Графички приказ λ–коцке

Правила извођења за λ–коцку су дата на табели 2.7. Као што се може видети λ–коцка има 8 темена и свако теме се може добити тако што се изабере различит скуп правила, тј. $s \in \mathcal{R}_1 = \{*, \Box\}$, а $(s1, s2) \in \mathcal{R}_2 = \{(*,*), (*,\Box), (\Box,*), (\Box,\Box)\}$:

$$\frac{}{\vdash * : \Box}\ \text{аксиома}$$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}\ \text{старт} \qquad \frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}\ \text{слабљење}$$

$$\frac{\Gamma \vdash F : (\Pi x : A.B) \qquad \Gamma \vdash \alpha : A}{\Gamma \vdash Fa : B[x := a]}\ \text{примена}$$

$$\frac{\Gamma, x : A \vdash b : B \qquad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash \lambda x : A.b : \Pi x : A.B}\ \text{апстракција}$$

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_2}\ \text{производ}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}\ \text{конверзија}$$

Табела 2.7: Правила закључивања λ–коцке

Избором различитих вредности скупова $\mathcal{R}_1$ и $\mathcal{R}_2$ добијају се темена λ–коцке:

$\lambda^{\rightarrow}$	$(*,*)$			
$\lambda 2$	$(*,*)$	$(\Box,*)$		
$\lambda\underline{\omega}$	$(*,*)$		$(\Box,\Box)$	
$\lambda\omega$	$(*,*)$	$(\Box,*)$	$(\Box,\Box)$	
λP	$(*,*)$			$(*,\Box)$
$\lambda P2$	$(*,*)$	$(\Box,*)$		$(*,\Box)$
$\lambda P\underline{\omega}$	$(*,*)$		$(\Box,\Box)$	$(*,\Box)$
$\lambda P2\omega = \lambda C$	$(*,*)$	$(\Box,*)$	$(\Box,\Box)$	$(*,\Box)$

При томе, уједно се може приметити да на основу правила (и одговарајућег избора) се могу креирати одговарајуће зависности и то:

$(*,*)$: термови зависе од термова
$(*,\Box)$: термови зависе од типова
$(\Box,*)$: типови зависе од термова
$(\Box,\Box)$: типови зависе од типова

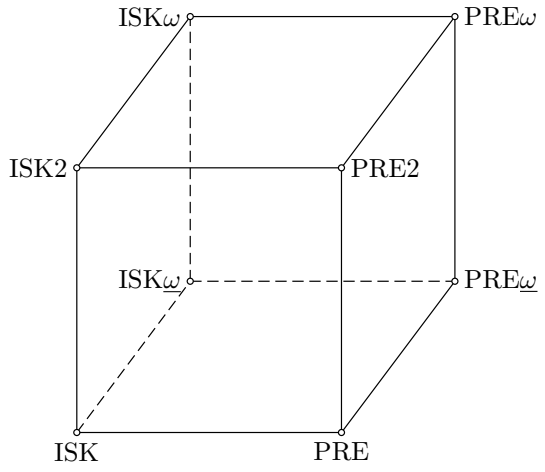
Сви елементи λ–коцке деле неке лепе особине, као што су строга нормализација и редукција субјекта. Сви системи имају одлучив проблем провере типа, док неки једноставнији системи (као што су $\lambda^{\rightarrow}$ и $\lambda 2$) имају одлучиво извођење типа.

У Кари–Хауард изоморфизму, λC одговара предикатској логици вишег реда. На пример, $\forall \equiv \lambda A : *.\lambda P : A \rightarrow *.\Pi a : A.Pa$ се може дефинисати и

има улогу универзалног квантификатора — узима тип и предикат тог типа и враћа предикат који одговара исказу да предикат важи за сваки елемент датог типа.

Систем λ–коцке има као везнике једино импликацију и универзални кватификатор. Могуће је направити проширења и увести нове типове тако да се формирају оператори који одговарају осталим везницима који постоје у интуиционистичкој логици. На пример, у систему који је направио Мартин–Леф [116], функцијском типу $\rightarrow$ одговара импликација, тип производа $A \times B$ одговара коњукцији, типови дисјунктне уније $A + B$ одговарају дисјункцији, зависни типови производа П–типови одговарају универзалном квантитфикатору, а зависни типови суме $\sum$–типови одговарају егзистенцијалном квантификатору.

Графички, на слици 2.2 коришћењем логичке коцке, ћемо приказати логику која одговара (преко Кари–Хауард изоморфизма) λ–коцки.



Слика 2.2: Графички приказ логичке коцке

Скраћенице које су коришћене на слици означавају:

ISK	Исказна логика
ISK$\underline{\omega}$	Слаба исказна логика вишег реда
PRE	Предикатска логика
PRE$\underline{\omega}$	Слаба предикатска логика вишег реда
ISK2	Исказна логика другог реда
ISKω	Исказна логика вишег реда
PRE2	Предикатска логика другог реда
PREω	Предикатска логика вишег реда

2.2 Кратак историјски преглед и осврт на главне карактеристике различитих интерактивних доказивача теорема

Главне карактеристике интерактивних доказивача теорема

Идеја иза интерактивног доказивања теорема је да се помоћу специјализованих софтверских алатки („*асистената за доказивање теорема*”) омогући корисницима да запишу довољно информација и смерница тако да систем може да потврди постојање формалног аксиоматског доказа. Заправо, корисник записује скице доказа, које систем проверава и истовремено рачунар аутоматски конструише комплетан доказ састављен од ситних корака. Асистенти за доказивање теорема се развијају преко 40 година и данас помоћу њих могу да се формализују велики делови разних математичких теорија. Интерактивне доказиваче теорема данас углавном користе обучени специјалисти који имају добро и математичко и програмерско знање. Поступак формализације је прилично тежак јер и када постоји математички доказ на папиру, потребно је додати још пуно детаља да би тај доказ био прихваћен од стране машине, тј. асистента за доказивање теорема.

Улога доказа је двојака. Прво, он мора да *убеди* читаоца да је тврђење коректно. Рачунар може много помоћи у овом делу јер је могуће написати програм који механички проверава исправност доказа. Потом, доказ мора и да *објасни* зашто је нешто коректно. Савремени асистенти за доказивање теорема имају и ову другу улогу доказа јер је могуће доказ записати у

облику који је читљив човеку. Можемо рећи да је главна замерка интерактивним доказивачима теорема што они замарају читаоца са превише детаља у доказу. Ипак, некада су управо ти детаљи веома битни. Додатно, често се аутоматизацијом успешно „крију" ситни, неинтересантни кораци.

Приступи у конструкцији доказивача теорема

Постоји много приступа како се механичка провера доказа може учинити сигурном и већина ових приступа се симултано користи у интерактивним доказивачима теорема. Овде ћемо навести четири основна приступа, а више информација се може погледати у [56].

- **Независан опис логике.** Потребно је да опис логичког система и његових математичких својстава (као што су механизми за дефинисање функција и типова података) буде независан од самог асистента за доказивање теорема. То значи да верујемо логици. Ово је основно својство које интерактивни доказивачи теорема морају да задовоље и већина других својстава заправо подразумевају постајање овог приступа.

- **Мало језгро.** Неки системи за доказивање теорема имају веома мало језгро са правилима која су таква да их корисник може верификовати ручном провером кода. Правила најчешће директно одговарају правилима одговарајућег логичког система. Сва друга правила у доказима су дефинисана на основу језгра, па је сваки корак доказа заправо скуп основних правила доказивања који се налазе у језгру. У овом случају потребно је да се верује *само* језгру.

- **Верификација система за проверу доказа.** Асистент за доказивање теорема је заправо само још један програм па је потребно и њега проверити, тј. формално верификовати, а то заправо значи да га треба записати у терминима логике и показати да је неко тврђење могуће показати у њему ако и само ако је то тврђење могуће извести коришћењем правила логике. Често, провера система се ради тако што се докаже да су све тактике (делови програма који праве кораке у доказима) које се користе у доказима исправне у логици и да за свако правило закључивања у логици (елементарно, основно правило) постоји тактика. Уколико програм већ има мало језгро онда је провера система олакшана јер је потребно верификовати само то језгро.

- **Де Брујинов критеријум.** (енг. *De Bruijn criterion.*) Интерактивни доказивач теорема конструише формалан доказ (*објекат доказа*), који је комплексан скуп података који потом проверава независни верификатор. То значи да исправност читавог система заправо зависи од тог независног верификатора (који је најчешће прилично једноставан). Многи асистенти за доказивање теорема генеришу и експлицитно чувају објекте–доказе. Као што смо видели, занимљива последица изоморфизма је да *провера доказа одговара провери типа*. Тиме у основи Де Брујиновог критеријума је Кари–Хауард изоморфизам о ком је било речи раније. Де Брујин (хол. *de Bruijn*) је искористио ову идеју за проверу, односно да се провера доказа заправо изврши алгоритмом за проверу типа.

Тактике

Асистенти за доказивање теорема користе доказивање уназад, односно кориснику приказују тврђење у облику циља који је потребно направити. Корисник на те циљеве примењује *тактике*, аутоматске алатке чији је циљ да докажу или упросте тврђење (сведу га на низ једноставнијих подциљева). Тактике морају да врате и оправдање како се од добијених подциљева изводи полазни циљ применом логичких правила и та оправдања се користе приликом провере крајњег доказа. Корисник може додавати тактике и проширивати могућности постојећих тактика. Такође, тактике могу звати друге тактике.

Приликом доказивања, тактика проверава стање доказа (скуп тренутних подциљева) и обавештава да није могуће применити тактику на тренутно стање или мења тренутно стање доказа. Односно, за дато стање тактика може имати три могућа излаза:

- успех: циљ је доказан,

- промена: тактика је направила неку промену тренутног стања и

- неуспех: тактика не може да докаже циљ и не може да направи промену тренутног стања доказа.

Декларативни наспрам процедуралног језика

Језик којим се записују докази у асистенту за доказивање теорема може бити **декларативни** или **процедурални**.

Поставља питање чему доказ служи — да ли желимо да човек чита тај доказ, да га разуме и мења или то треба да буде само скуп инструкција намењен за извршавање на рачунару. Заправо, улога доказа је двојака, да *оправда* тврђење и да *објасни* зашто тврђење важи.

Доказ записан у процедуранлом стилу је „нечитљив" јер корисник не може да види кораке у доказивању и овај доказ једино има смисла када се он извршава у одговарајућем доказивачу теорема. Процедурални језик је дизајниран са намером да *оправда тврђење*, уз њих често иде моћна аутоматизација доказивања и често су докази концизни и кратки.

У декларативном стилу доказ се записује тако да се могу уочити кораци у доказивању без извршавања у доказивачу теорема и математичари лако могу да уоче какво је резоновање коришћено у доказу. Декларативни језици конструисани су са намером да *објасне* тврђења. Ипак декларативни докази су често значајно дужи. Иако је ово важно за онога ко чита доказ, онај ко пише доказ има значајно више посла.

Можемо упоредити два приступа на примеру доказа Пирсовог закона (једноставна исказна таутологија). Докази су записани у систему *Isabelle/HOL*, при чему је леви доказ дат у декларативном стилу, а десни доказ у процедуралном стилу. Како ћемо касније детаљније објашњавати *Isabelle*, сада нећемо улазити у објашњавање синтаксе, већ ћемо се пре свега концентрисати на разлике између процедералног и декларативног стила.

```
lemma"((A ⟶ B) ⟶ A) ⟶ A"
proof
   assume "(A ⟶ B) ⟶ A"
   show "A"
   proof(rule classical)
     assume "¬A"
     have "(A ⟶ B)"
     proof
        assume "A"
        with `¬A` show "B"
        by contradiction
```

```
lemma "((A ⟶ B) ⟶ A) ⟶ A"
apply (rule impI)+
apply (rule classical)
apply (erule impE)
apply (rule impI)
apply (erule notE, assumption)+
done
```

```
    qed
    with `(A ⟶ B) ⟶ A` show A ..
  qed
qed
```

У десном доказу, сваки корак доказа представља примену неког правила природне дедукције. Ипак, овако записано, није могуће видети стање доказа и тек када се покрене доказивач може се видети ефекат примене правила на дату формулу. На пример, након примене `impE` правила стање доказа је:

```
goal (2 subgoals):
1. ¬A ⟹ A ⟶ B
2. ⟦¬A; A⟧ ⟹ A
```

Са друге стране, код левог, декларативног доказа можемо видети експлицитне међукораке у доказу. Иако је овакав доказ дужи, он се може анализирати и разумети без покретања у неком доказивачу теорема. Заправо, и у овом доказу се примењују правила природне дедукције. Команда `proof` у овом примеру одмах примени правило за импликацију, те тако можемо претпоставити (`assume`) леву страну, а желимо да докажемо десну страну. Доказ се обично завршава позивом неке аутоматске тактике, ми смо овде користили `contradiction` (да би показали контрадикцију) и конструкт ". ." (аутоматски пролази одговарајуће правило природне дедукције и решава циљ).

Доказивање унапред и доказивање уназад

Доказивачи могу имати и различите приступе при доказивању, односно, постоји доказивање унапред и доказивање уназад.

Приликом *доказивања унапред* (енг. *forward proof*) корисник се труди да од задатих хипотеза изведе нове хипотезе, тако да на крају изведе и цињни закључак. Рецимо, ако су дате хипотезе $H = [H_1, \ldots, H_n]$, а треба показати циљ G, корисник у сваком кораку изводи из хипотеза нову хипотезу $\Phi(H)$ и додаје у скуп хипотеза, односно ново стање је $H = [H_1, \ldots, H_n, \Phi_1(H), \Phi_2(H), \ldots]$. Успехом се сматра када се циљ нађе међу хипотезама, односно када је $H = [H_1, \ldots, H_n, \Phi_1(H), \Phi_2(H), \ldots, G, \ldots]$.

Као што је било речи, код *доказивања уназад* (енг. *backward proof*) корисник задаје теорему која треба да се покаже у форми циља, а потом примењује *тактике* које трансформишу циљ у једноставније подциљеве. Подциљеви се

лако доказују или се даље раздвајају на мање подциљеве који могу бити једноставно доказани. То значи да тактике генеришу листу подциљева. Поново, посматрајмо малопређашњи пример дате хипотезе $H = [H_1, \ldots, H_n]$, а треба доказати циљ G. Из циља G се најпре изводе подциљеви које је потребно доказати, рецимо $G_1, \ldots, G_k$, а онда је циљ доказати да за сваки од k подциљева важи да се могу извести из хипотеза $H = [H_1, \ldots, H_n]$. Поред одређивања подциљева, тактике враћају функцију која оправдава и реконструише првобитни циљ након што су подциљеви доказани.

За поређење ова два приступа представићемо једноставан пример чији доказ је писан у систему *Isabelle/HOL*. При томе, леви доказ је пример доказивања унапред, а десни доказ је пример доказивања уназад.

```
lemma A ∧ B ⟶ B ∧ A
proof
   assume A ∧ B
   from A ∧ B have A ..
   from A ∧ B have B ..
   from A B have B ∧ A ..
qed
```

```
lemma A ∧ B ⟶ B ∧ A
proof
   assume A ∧ B
   show B ∧ A
   proof
      show B by (rule conjunct2) fact
      show A by (rule conjunct1) fact
   qed
qed
```

У пракси се ова два приступа најчешће симултано примењују. Доказивачи чији докази су писани у декларативном стилу често комбинују доказивање унапред и уназад, док у процедуралном стилу фаворизују доказивање уназад.

Кратак историјски осврт

Занимљиво је напоменути да се око 1970. године на више места истовремено створила идеја машински проверивих доказа и тада су настали први системи од којих су неки били веома утицајни на развој савремених доказивача теорема. Више о историјском прегледу интерактивних доказивача теорема се може пронаћи у радовима [56, 112, 8, 66].

Рана историја

Један од најстаријих система за доказивање теорема је Де Брујинов **AutoMath** [44], који се појавио крајем 1960. Он је имао малу моћ верификовања доказа јер је најзначајнија идеја у пројекту била развијање компактне, ефикасне нотације за опис математичког доказа. Идеја је била развити математички језик којим би се сва математика могла прецизно записати, али у смислу да лингвистичка тачност повлачи математичку тачност. Језик система је био такав да се оно што је записано у њему може проверити помоћу рачунара, али помоћ рачунара у конструисању доказа је била минимална. Систем је имао мало језгро и био је прилично једноставан. У систему *AutoMath* се први пут јавила идеја да се доказ представи као објекат у неком формалном језику (Де Брујинов критеријум). Многи системи који су се касније развили заснивали су се на овој идеји — *LF* [73], *Lego* [107], *Alf* [110], *Agda* [3], *Coq* [39] и *NuPrl* [38]. Додатно, треба нагласити да је *AutoMath* био систем за проверу доказа, али не и асистент за доказивање теорема. Корисник би написао терм доказа и систем би проверио његов тип. То је другачије у односу на асистенте за доказивање теорема где корисник пише тактике које интерактивно наводе систем како да конструише терм доказа. Још једна важна идеја која се појавила у пројекту *AutoMath* је **логички оквир**. Де Брујин је инсистирао на идеји да систем само омогући основне математичке механизме супституције, креирања и развијања дефиниција и слично, а да корисник додаје логичка правила која жели.

Следећа прекретница је Јутингова (енг. *Jutting*) докторска теза из 1977. године [89]. Поред ограничења која су имали тадашњи рачунари и мањка софтверске подршке, он је коришћењем система *AutoMath* успешно представио комплетну формализацију Ландоове (нем. *Landau*) књиге „Основе анализе" (нем. *„Grundlagen der Analysis"*) [99], где је формализована конструкција реалних бројева преко Дедекиндових пресека и закључено да реални бројеви који су конструисани на такав начин формирају комплетно уређено поље. Након тога, доказивачи теорема се мењају и у њих се уграђује додатна подршка за израчунавања.

Доказивачи засновани на модерној теорији типова

Још једна прекретница у развоју интерактивних доказавача теорема било је откриће Кари-Хауард изоморфизма који даје везу између природне дедукције и типизираног λ-рачуна. Ово откриће је служило као основа за **модерну теорију типова** која је основа многим савременим интерактивним доказивачима теорема. Поред једноставних типова, овај рачун је омогућио постојање зависних типова (енг. *dependent types*) (типова који зависе од вредности) и то је омогућило да се многе логичке претпоставке кодирају као формуле тако да се испитивање исправности формуле своди на испитивање типа у одговарајућој теорији. **Мартин Лоф** (шве. *Martin-Löf*) је проширио ове идеје и развио је конструктивну теорију типова, где су индуктивни типови и функције дефинисане коришћењем добро–засноване рекурзије основни појмови [115, 132]. Током година, Мартин Лоф је развио више теорија. Прва је била имплементирана у асистенту за доказивање теорема *NuPrl* [38], касније је развио системе *ALF* [110] и *Agda* [3], а већина његових идеја је нашла пут до савремених асистената за доказивање теорема.

Најуспешнији интерактивни доказивач теорема (награђен 2013. године *ACM* софтверском системском наградом) који се заснива на модерној теорији типова је **Coq** и њега су развијали (са многим сарадницима) Кокан (фр. *Coquand*) и Хует (фр. *Huet*). Парадигма докази-као-програми је једна од кључних одлика система *Coq* [105]. Као последица Кари–Хауард изоморфизма, конструктиван доказ је изоморфан функционалном програму, и зато у систему *Coq* је из доказа могуће извести програме у реалном функционалном програмском језику (за сада доступни су излази у језицима *Haskell* [83], *Objective Caml* [104] и *Scheme* [154]). Овај механизам је важан јер се на тај начин могу добити верификовани програми.

Такође, могуће је програмирати функције као програме у оквиру система *Coq* јер систем садржи (мали) функционални језик са апстрактним типовима података. Једна од најважнијих примена јесте могућност да се имплементира алгоритам за проверу типа који као улазне податке прима контекст и терм, а као излазни податак даје тип (ако тип који задовољава улазне податке постоји) или враћа неуспех ако тип не постоји. О вези између терма и типа смо говорили раније, у оквиру Кари–Хауард изоморфизма. Да би се додатно проверила поузданост система *Coq*, у оквиру пројекта *„Coq in Coq"* [11] урађена је верификација овог алгоритма за проверу типа у оквиру самог система *Coq*.

Верификовано је следеће тврђење:

$$\Gamma \vdash M : \tau \Leftrightarrow TC(\Gamma, M) = \tau$$

при чему је TC алгоритам за проверу типа, Γ је контекст, M је терм, а τ је тип. Ово није могуће доказати без претпоставке да су сви термови строго нормализовани, што следи из својства да су све функције које се могу дефинисати у систему *Coq* тоталне, што на мета нивоу може се доказати за систем *Coq*. Програм TC је имплементиран у систему *Coq*, али се може екстраховати и користити као проверавач типа за нове верзије система *Coq*.

Систем *Coq* се и даље развија. Многе важне теорије су показане у систему *Coq*: теорема о обојености графа са четири боје [61], основна теорема алгебре [57], теорема о простим бројевима [63], формално верификован компилатор [102].

PVS (Prototype Verification System) [134] се развија од 1992. године. Циљ система је да комбинује предности потпуно аутоматских доказивача теорема (које имају моћне процедуре за одлучивање, али малу изражајност) са предностима интерактивних доказивача теорема које имају много изражајнији језик и логику. Заснива се на теорији типова и има типизирану логику вишег реда. Логика система *PVS* није независно задата и систем нема мало језгро или објекте који могу бити независно проверени. Зато се са времена на време деси да се пронађе нека неконзистетност у систему која се онда поправља. Цена аутоматског закључивања је да понекад доказ оде у нежељеном правцу. Користио се у неколико истраживања исправности софтвера који има индустријску примену, као што је, на пример, софтвер за аутоматско организовање реда летења [126]. Често служи као алатка за верификацију рачунарске алгебре и користи се у системима за верификацију кода.

Неки од модерних система који су засновани на теорији типова су *Matita* 5 [5], *Agda* 6 [133] и *Epigram* 7 [118].

Доказивачи засновани на *LCF* приступу

Већина система за интерактивно доказивање теорема се заснива на архитектури коју је развио Милнер (енг. *Milner*) 1972. године, **LCF проверавач доказа** [123], који је заправо имплементација логике израчунљивих функција коју је развијао Скот (енг. *Scott*). *LCF* је скраћено од „Логика израчунљивих функција” (енг. *„Logic for Computable Functions”*). То је предикатска логика

над термовима једноставног λ-рачуна са типовима. *LCF* је био погодан за резоновање о семантици програма и о израчунљивим функцијама над целим бројевима, листама и сличним доменима.

Да би омогућио да се безбедно додају нове команде за доказ, али да се докази не чувају у меморији (већ само чињенице које су доказане), Милнер је унапредио и развио систем и то је данас познато као **LCF принцип**. Доказивач у *LCF* стилу (који задовољава *LCF* принцип) је заснован на малом, основном језгру кода, којем се верује, и у ком су имплементиране основне аксиоме логике и основна правила логике и који служи за извођење теорема применом тих основних правила аксиоматског система за логику за коју је доказивач у *LCF* стилу имплементиран. Такав систем може садржати и нешто изражајније делове (а сви су изграђени над овим малим језгром) који омогућавају имплементацију комплекснијих процедура за доказе које се могу рашчланити заправо у много позива основних правила. Исправност се гарантује тиме што само основна правила могу мењати стања у доказима јер све што систем ради зависи од основног језгра којем се верује. Оваква ограниченост се углавном постиже коришћењем функционалних програмских језика као што су *ML* [74] и *OCaml* [100] и применом основних правила извођења као јединих могућих конструктора апстрактног типа.

Да би био испуњен де Брујинов критеријум потребно је чувати објекте доказа, а то заузима много меморије. Други проблем који је уочен је фиксан скуп доказ–команди (дата и изведена правила и процедуре за аутоматско доказивање) који није било могуће лако проширити. Милнер је дошао на идеју да се чувају само теореме, али не и њихови докази. Односно, кораци доказа би били изведени, али не би били памћени. Ипак, да би осигурао да се теореме могу добити само доказом, Милнер је дошао на идеју коришћења *апстракног типа података*. Апстрактан тип података за запис теорема, тј. апстрактан тип теорема, најчешће означаван са `thm` (скраћеница од *theorem* (срб. теорема)) је такав да су једине константе овог типа (предефинисане вредности) аксиоме, а једине функције (операције) над овим типом су правила закључивања. Коришћењем апстрактног типа је постигнуто да систем има веома мало основно језгро – тип `thm` и његове конструкторе. Строга провера типа је осигурала да једине вредности које се могу креирати су оне које се могу добити из аксиома применом низа правила закључивања (интересантно је напоменути да не постоје литерали типа `thm`). Овим приступом је избегнуто

експлицитно памћење доказа–објеката, а задржан је исти ниво поузданости.

Милнер је желео и да омогући да корисник прави своје, нове доказ–комаде. Зато је развио специјализовани програмски језик **ML** [74]. Језик је строго типизиран да би могао да да подршку за механизам апстрактног типа који је потребан за осигуравање исправности теорема. Строга провера типа језика *ML* осигурава да се ни једна теорема не може креирати а да није прошла (доказана) кроз мало фиксно језгро. То значи да су све теореме које постоје у систему вредности типа `thm` и доказиве су у оквиру логике за коју је систем имплементиран (јер се добијају коришћењем правила извођења из аксиома). Многе доказ–команде су углавном већ доступне у систему, али корисник може додати своје. Ово повећава могућности доказивача али не нарушава исправност система. Коректност система не зависи од коректности доказ–команди које су имплементиране у језику *ML* и које могу имати грешке већ корекност се заснива на чињеници да су добијене теореме вредности типа `thm` (односно све се своди на мало, исправно језгро).

Првобитно доказ је извођен тако што се кретало од главног циља, а онда се доказ делио на два подциља (коришћењем фиксног скупа команди за дељење у подциљеве, као што је, на пример, примена индукције) и ти подциљеви су разрешавани симплификатором или даљем дељењем на подциљеве. Овакав приступ одговара *доказивању уназад*. Милнеров приступ је увео могућност *доказивања унапред*, односно од апстрактног типа ка теореми. Било је потребно направити алате који би били оријентисани ка циљу, па је језик *ML* направљен да буде функционалан. Стратегије за прављење подциљева (Милнер их је назвао *тактике*) су биле програмиране као функције, а операције за комбинацију стратегија су програмиране као функције вишег реда којима се прослеђује стратегија и које враћају стратегију. Предвиђено је да се може десити ситуација да примена тактике не успе (на пример, када је тактика примењена на лош циљ) и зато је развијен механизам изузетка који сигнализира када је правило (односно) тактика лоше примењена.

Поред Милнера на овом приступу су радили Полсон (енг. *Paulson*) и Хует који су доста унапредили имплементацију и дизајн [138]. Они су увели комплетну предикатску логику и додали обиман скуп тактика и имплементирали напредне алатке за доказивање (као што су симплификатор и презаписивање).

Пратећи *LCF* принцип, а имајући на уму верификацију хардвера, Гордон (енг. *Gordon*) је имплементирао **HOL** [125]. Систем је отвореног типа и мно-

го истраживача је допринело његовом развоју. Утицао је на развој других система, као што су *Isabelle/HOL* и *HOL Light*. **HOL Light** [77, 75] је развијао Харисон (енг. *Harrison*) и он има једноставнију логичку основу него други *HOL* системи. Додатно, има велику библиотеку математичких теорема (нпр. аритметика, скупови, реална анализа итд.). Треба још напоменути да су идеје *LCF* принципа примењене и у систему *Coq*.

Током 1980-их развијено је много специјализованих логика и за сваку од њих било је потребно имплементирати доказивач у *LCF* стилу. Прављење доказивача испочетка за сваку нову логику постало је изазовно и захтевно. Зато, развили су се **генерички доказивачи** који имају *мета-језик* и *мета-логику* што омогућава формализацију у различитим објектним логикама. Правила доказивања су описана декларативно (а не као *ML* програми).

Полсон је 1986. године развио **Isabelle** [136, 137], генерички доказивач теорема који је имплементирао многе теорије (нпр. интуиционистичка природна дедукција, теорија конструктивних типова, класична логика првог реда итд.). Данас је најраширенији **Isabelle/HOL** [131], интерактивни доказивач теорема заснован на логици вишег реда.

Остали значајни доказивачи

Mizar [158], систем који је Трибулек (пољ. *Trybulec*) представио 1973. године и који је још у употреби, користи аутоматске методе да провери формалне доказе написане у језику који је конструисан на такав начин да симулира неформалан математички жаргон. *Mizar* пројекат је најдуже у континуитету одржаван и развијан. Могло би се рећи да постоје два *Mizar* пројекта: *Mizar језик* чији је циљ да буде формални језик близак математичком језику и *Mizar систем*, који је рачунарски програм који проверава математичку исправност текстуалних докумената записаних у језику *Mizar*. Овај систем се заснива на Тарски–Гротендик (енг. *Tarski–Grothendieck*) [157] теорији скупова са класичном логиком, а докази се записују у стилу Јанковског (пољ. *Jankowski*) [88], што је данас познатије као Фич стил (енг. *Fitch-style*) [48]. У почетку развоја пројекта нагласак је био више на едитовању и записивању математичких чланака него на проверавању доказа. Инсистирано је на креирању библиотеке формализоване математике која би заправо била један повезан скуп свих математичких знања и тренутно *Mizar* пројекат има највећи репозиторијум формализоване математике. Ипак, због ефикасности не подржава велике ра-

звоје и чланци се шаљу „комитету библиотеке" која га може укључити у *Mizar* библиотеку. Врло значајна идеја је запис текста на декларативном језику чији циљ је да приближи текст уобичајеном математичком језику. Ово додатно значи да језик има бројна специјална својства и конструкције које омогућавају уграђену аутоматизацију. Зато, систем нема мало језгро и не задовољава Де Брујиново својство.

Језик *Mizar* је инспирисао друге асистенте за доказивање теорема да развијају декларативни језик за доказе. Један од најзначајнијих је језик **Isar** [166] који је развио Венцел (нем. *Wenzel*). То је декларативни језик за *Isabelle* интерактивни доказивач теорема и доста се користи од стране Isabelle корисника. Поред *Isar*-а, Харисон је развио *Mizar-мод* за *HOL Light* [167], а Коберне (фр. *Corbineau*) је развио *C-zar* декларативни језик за *Coq* [40].

Nqthm [21] доказивач теорема, који се и данас користи, био је представљен почетком 1970. као потпуно аутоматски доказивач теорема, али се у међувремену идеја пројекта променила и наставило се са изградњом метода које омогућавају кориснику да тврђења доказује поступно тако што у корацима доказа записује потребне чињенице које аутоматски доказивач користи да би комплетно извео доказ. Овај систем је развијан у функционалном програмском језику *Lisp* [119]. За разлику од многих савремених система који користе логику вишег реда, логика у овом систему је логика првог реда без квантификатора са једнакошћу, прилично примитивном рекурзивном аритметиком што чини аутоматизацију прилично моћном, али изражајност је слаба. Идеја комбинације моћне аутоматизације са могућностима интерактивног доказивања је пронашла пут и до многих других доказивача теорема. Корисник интерактивно ради у систему тако што додаје нове леме које су заправо кораци у доказивању жељене теореме, систем потом покушава да докаже леме аутоматски, а онда покушава да докаже теорему коришћењем датих лема. Систем *Nqthm* је еволуирао у *ACL2* (*A Computational Logic for Applicative Common Lisp*) [143], интерактивни доказивач теорема који се доста користи у индустрији.

2.3 Важни резултати и пројекти у области интерактивног доказивања теорема

Најпре ћемо се осврнути на неколико радова из формализације математике који су важни и због своје улоге у формализацији геометрије (бројни резултати су коришћени као познате чињенице приликом формализације геометрије), али и због свеукупног значаја и утицаја на формализацију и интерактивно доказивање теорема.

Формално су доказане Брауерова теорема фиксне тачке [80], основна теорема алгебре [122, 57], Геделова теорема непотпуности [151], многе теореме реалне анализа [79, 42]. У најзначајније постигнуте резултате до данас могу се убројати и формализација теореме о простим бројевима [7], затим формални доказ о обојивости графа са четири боје [62].

Важно је поменути и актуелне пројекте великих размера чији циљ обухвата формализацију великих делова математике и у којима учествују многи научници. У оквиру пројекта чији руководилац је Гонтије (фра. *Gonthier*) је успешно формално доказана Фејт–Томпсонова (енг. *Feit–Thompson*) теорема (која се још и назива теорема непарног реда) [64]. Формализација је рађена уз помоћ асистента за доказивање теорема *Coq* и језика за доказе *SSreflect* [65]. Ова теорема је била веома важан корак у *класификацији коначних једноставних група*. Оригинални доказ на папиру је заузимао 225 страна, док формализација има 150000 линија кода, 4000 дефиниција и 13000 лема и теорема. Да би могли да формализују ову теорему, аутори су морали да формализују и бројна тврђења и својства линеарне алгебре, теорије коначних група, теорије Галоа. Наилазили су на бројне празнине и грешке, од којих неке није било лако исправити и допунити.

Други важан пројекат је пројекат *Flyspeck* [72] који је покренуо Хејлс (енг. *Hales*) да би могао формално да докаже Кеплерову хипотезу: „Најгушће паковање једнаких сфера је паковање у тесерални кубични кристални систем". У оквиру овог пројекта формално је показано много математичких тврђења, направљена је велика база математичког знања која може да послужи у неким новим формализацијама.

Поред формализације математике, коришћењем асистената за доказивање теорема рађена је и *верификација софтвера*. Значајан резултат је *CompCert* [101, 103], формално верификован компилатор за програмски језик *C* и *L4*

[94, 93], формално верификован оперативни систем.

Потенцијалне грешке у формално верификованим доказима

У раду из 2016. године ([2]) приказује се начин како би могла да се изврши *ревизија формализације*. Наиме, аутори сматрају да иако формализација даје стриктан и прецизан приступ математици, и даље су могуће неке грешке. Једна од грешака која се често спомиње је да се формализовано тврђење разликује од тврђења за које је рађена формализација или од тврђења за које мислимо да је показано. Ова грешка може настати због лоших дефиниција које се могу провлачити кроз целу формализацију и тако утицати на крајњи исход формализације. Поред ових грешака, аутори истичу и неке недостатке најпопуларнијих доказивача теорема, а међу недостацима се посебно истиче комплексност система који се користи и који може бити место потенцијалним грешкама у самом доказивачу. Њихов приступ ревизији се своди на неколико корака, при чему није циљ проверавати сваку линију комплетне формализације, већ завршно стање формализације. Такође, предлажу да ако је формализација рађена у једном језику, да се ревизија ради у другом (или у више других) језика и дају пример неколико алатки које врше превођење из једног језика у други. Коначно, дају пример како је могуће извршити ревизију над делом *Flyspeck* пројекта.

2.4 Isabelle/HOL

Isabelle [136, 137] је генерички асистент за доказивање теорема, има бројне специјализације за различите логике, а најразвијенија је за логику вишег реда, *Isabelle/HOL* [131]. Углавном је прилагођен свакодневној математичкој нотацији. На развој овог система је радило (и још увек ради) пуно научника, али најзначајнији аутори су Лари Полсон, Нипков (нем. *Nipkow*) и Венцел. Као што смо раније напоменули, на развој *Isabelle/HOL* система је значајно утицао *LCF* принцип. Систем *Isabelle/HOL* је заснован на малом језгру на основу кога је све даље развијано. Поступак формализације математичких теорија се састоји из дефиниција нових појмова (типови, константе, функције и др.), и доказивања тврђења која за њих важе (леме, теореме и др.). *Isabel-*

le/HOL има обимну библиотеку теорија која се стално увећава. Велики број формализација из математике и рачунарства је доступан кроз Архив формалних доказа (енг. *Archive of Formal Proofs*)[13]. У овом раду ми смо користили неке од тих библиотека.

Рад у систему *Isabelle/HOL* подразумева креирање *теорија* — именоване колекције типова, објеката, функција, теорема и лема. Општи формат теорије `T` је:

```
theory T
imports B1 B2 ... Bn begin
deklaracije, definicije, dokazi
end
```

при чему су $\texttt{B}_1$ $\texttt{B}_2$ ... $\texttt{B}_n$ неке већ постојеће теорије, а `deklaracije, definicije i dokazi` су нови концепти које задаје корисник. Све што је дефинисано и доказано у $\texttt{B}_1$ $\texttt{B}_2$ ... $\texttt{B}_n$ је видљиво и у теорији `T` и може се користити у дефинисању нових појмова и у доказима тврђења. Теореме и леме се користе да изразе својства и карактеристике дефинисаних објеката и функција. Након сваке теореме или леме се налази и њен доказ исправности. Доказе које задаје корисник систем аутоматски, формално верификује.

Типови, термови, променљиве

HOL је типизирана логика. Типови су екстремно важни и систем *Isabelle* инсистира да све формуле и термови морају бити добро типизирани, а у супротном ће обавестити да је дошло до грешке. У систему *Isabelle/HOL* имамо неколико врста типова које ћемо овде укратко представити.

Основни типови

Isabelle/HOL има предефинисане типове, као што су, на пример, тип `bool` који означава тип за вредности тачно и нетачно, `nat` означава природне бројеве, тип `int` означава целе бројеве, `real` означава реалне бројеве, док тип `complex` означава комплексне бројеве, а `rat` рационалне бројеве.

Тип природних бројева је изграђен коришћењем конструктора 0 и `Suc`. Предефинисане су основне аритметичке операције (нпр. $+$, $-$, $*$, $/$, `div`, `mod`),

[13] `http://afp.sourceforge.net/`

као и релације $<$ и $\leq$. Иначе, аритметичке операције су преоптерећене јер се могу примењивати и на друге типове, а не само за природне бројеве.

Имагинарна јединица се обележава са ii. Конверзија из реалног у комплексан број се означава са `cor`, реални и имагинарни део комплексног броја са `Re` и `Im`, комплексни коњугат са `cnj`, модуо комплексног броја са $|\_|$, и аргумент комплексног броја са `arg` (у систему *Isabelle/HOL* он је увек у интервалу $(-\pi, \pi]$). Комплексна функција за знак `sgn` одређује комплексан број на јединичној кружници који има исти аргумент као и дати ненула комплексан број (нпр. `sgn` $z = z/|z|$). Ова функција је преоптерећена и такође се примењује и за реалне бројеве (преоптерећење је математички оправдано у овом случају јер важи `sgn` (x + ii*0) = `sgn` x). Функција `cis` примењена на α израчунава `cos` α + ii∗`sin` α.

Типске променљиве

Типске променљиве користимо када не знамо ког типа је нека вредност. Коришћењем ограничења које даје израз у коме се појављује вредност непознатог типа, *Isabelle/HOL* решава израз за све вредности за које су та ограничења испуњена. Означавамо са апострофом и најчешће малим словом, на пример, `'a`, `'b` итд. Ове типске променљиве су посебно значајне јер се помоћу њих могу изградити полиморфни типови. На пример, тип `'a` ⇒ `'b` представља тип функције идентитета.

Конструктори типа

Постоји много конструктора типа, а нама су најинтересантнији конструктори за листе и скупове.

Конструктор за листу је `list`. Листа над типом `'a` се означава са `'a list`. Израз x`#`y означава листу са главом x, и репом y. Израз x`@`y означава спајање две листе, x и y.

Скуп елемената типа `'a` се означава са `'a set`. *Isabelle/HOL* скуповна теорија је веома слична скуповној теорији у стандардној математици, са неколико мањих изузетака. Разлика скупова се означава са $X - Y$, а слика функције f над скупом X се записује са f‘X. Тип производа је означен са $\tau_1 \times \tau_2$ (где су τ_1 и τ_2 типови скупова).

Функцијски типови

Тип функције се означава као $\tau_1 \Rightarrow \tau_2$. Функције су углавном записане у Каријевом облику, а примена функције се углавном записује у префиксној форми, што је често случај код функционалног програмирања, као `f` x (уместо $f(x)$, што је ближе стандардној математичкој нотацији). *Isabelle/HOL* подржава и скраћену нотацију $[\![\tau_1, \ldots, \tau_n]\!] \Rightarrow \tau$ је скраћеница за $\tau_1 \Rightarrow \ldots \Rightarrow \tau_n \Rightarrow \tau$ (*Isabelle/HOL* посматра $\tau_1 \Rightarrow \tau_2 \Rightarrow \tau_3$ као $\tau_1 \Rightarrow (\tau_2 \Rightarrow \tau_3)$ и имајући то на уму ми у претходном изразу нисмо писали заграде).

Предикат `inj` означава да је функција инјективна, `bij` да је бијекција, а `continuous_on` да је непрекидна на датом скупу (претпостављајући да је одговарајући тип метрички простор за неку дату функцију растојања).

Термови

Термови се праве по узору на термове λ–рачуна са мало проширеном синтаксом.

Применом функција на аргументе добијају се термови. Ако је `f` функција типа $\tau_1 \Rightarrow \tau_2$, а `t` терм типа τ_1, онда је `f t` терм типа τ_2.

За конструкцију термова могу да се користе користе *let–конструкције*: **`let`** $x = t$ **`in`** u што је еквивалетно са u у коме су сва слободна појављивања променљиве x замењена са t (нпр. **`let`** $x = 3$ **`in`** $3 * x$ је еквивалентно са $3 \cdot 3$). Могуће је направити и комплекснију *let–конструкције*: **`let`** $x_1 = t_1$`;` $x_2 = t_2$`;` $\ldots$ $x_n = t_n$ **`in`** `u`.

Такође, могуће је користити *if–then–else изразе*: **`if`** b **`then`** t_1 **`else`** t_2 (нпр. **`if`** $x > 0$ **`then`** x **`else`** $-x$). Тип за b мора бити `bool`, а t_1 и t_2 морају бити истог типа.

Термови се могу градити и коришћењем *case–израза*: **`case`** e **`of`** $c_1 \Rightarrow e_1$ `|` $c_2 \Rightarrow e_2$ `|` …`|` $c_n \Rightarrow e_n$ што има вредност e_i ако је e једнак неком c_i.

Подржано је и коришћење неких инфиксних функција (на пример, $+$).

Термови су λ–изрази, те могу бити облика $\lambda x.$ `f` x. На пример, $\lambda x.x + 5$ је функција са једним аргументом, x, а враћа $x + 5$.

И поред подржаног механизма дедукције типова, некада је потребно експлицитно навести тип терма. Да би изразили да је неки терм t типа τ пишемо $t :: \tau$.

Логичке формуле су записане у логици вишег реда коришћењем стандардне нотације, везници су $\wedge$, $\vee$, $\neg$, $\longrightarrow$, а квантификатори су $\forall$ и $\exists$. Постоји и $\exists!x.f$ што означава да постоји тачно једно x такво да задовољава P.

Променљиве

У систему *Isabelle/HOL* постоје везане и слободне променљиве, али и схематске или непознате променљиве које се означавају са ?*ime_promenljive*. Логички гледано, то су заправо слободне променљиве, али се може десити да буду иницијализоване од стране неког терма током процеса доказивања.

Увођење нових типова

Нов тип се може увести на неколико различитих начина и овде ће бити укратко представљени.

Најједноставнији начин је да се користи команда **type_synonym** која уводи ново име за већ постојеће типове.

Алгебарске типове, као и рекурзивне типове података је могуће увести коришћењем кључне речи **datatype**. Следи неколико једноставних примера.

```
(1) datatype accuracy = True | False | Maybe
(2) datatype 'a option= None | Some 'a
(3) datatype nat      = 0 | Suc nat
(4) datatype 'a Tree  = NIL | Node "'a Tree" "'a" "'a Tree"
(5) datatype even_nat = 0 | Suc odd_nat
    and      odd_nat  = Suc even_nat
```

Прва два типа нису рекурзивна, при чему је тип `accuracy` једноставан, набројиви тип, а тип `option` је полиморфни тип. Други и трећи тип су једноставни рекурзивни типови којима се дефинишу природни бројеви и стабло. Коначно, последња два типа (`even_nat` и `odd_nat`) представљају примере узајамно рекурзивних типова. Након дефинисања типа добијају се функције које називамо конструкторима тог типа (нпр. `Node` је конструктор који прима лево подрво, елемент у корену, десно подрво и враћа дрво изграђено од тих елемената).

Често се приликом рада са неким термовима типа који је дефинисан са **datatype** користе *case–изрази* јер се помоћу њих једноставно могу анализи-

рати елементи тог типа. Приликом дефинисања функција за рад над неким типом дефинисаним са **datatype** често се користи конструкција са **primrec**.

```
primrec left :: "'a Tree ⇒ 'a Tree" where
    "left (Node l v r) = l"
```

Са **primrec** су функције задате примитивном рекурзијом, односно са сваким рекурзивним позивом издваја се конструктор типа за један од аргумената. То имплицира да се рекурзија сигурно завршава и да је функција тотална. Постоје и други начини да се задају функције над рекурзивним типом, рецимо коришћењем **fun** или **function**, али о томе ће бити више речи касније.

Сваки нови тип је спецификован да буде изоморфан са неким непразним подскупом постојећег типа. На пример, тип се може увести као `typedef three = "{0::nat, 1, 2}"` , а то генерише обавезу да се докаже да је тип непразан. Бијекција између новог апстрактног типа и његове репрезентације дата је са две функције: `Rep_three :: three ⟹ nat`, и `Abs_three :: nat ⟹ three`, које задовољавају услове `Rep_three` $x \in \{0, 1, 2\}$, `Rep_three (Abs_three` x`) =` x, and $y \in \{0, 1, 2\} \implies$ `Abs_three (Rep_three` y`) =` y.

Записивање дефиниција, тврђења и корака у доказивању

Дефиниције се задају коришћењем синтаксе **definition** x **where** "x = ...", где је x константа која се дефинише.

```
definition is_empty :: "'a Tree ⇒ bool" (infix "≈" 50) where
[simp]: "is_empty l ⟷ l = NIL"
```

Овом дефиницијом смо дефинисали функцију која испитује да ли је стабло празно. Дефиниција је на неки начин скраћеница – ново име за постојеће концепте. Дефиниција не може бити рекурзивна. Уколико желимо да неку дефиницију додамо у правила за симплификацију пишемо [simp]. Може се користити команда **infix** која омогућава да некој операцији доделимо инфиксну ознаку. Број означава приоритет тог инфиксног оператора и пракса је да уведени инфиксни оператори имају мањи приоритет него они већ постојећи, па се зато стављају велики бројеви.

Леме се задају коришћењем синтаксе

lemma *name*
  fixes *vars*
  assumes *assms*
  shows *concl*

при чему је *name* је име леме, *vars* услови који важе, *assms* претпоставка, а *concl* је закључак тврђења. Ако нема претпоставки, кључна реч **shows** може бити изостављена. Такође, користићемо синтаксу **lemma** „$\bigwedge x_1, \ldots x_k.\ [\![asm_1; \ldots; asm_n]\!] \Longrightarrow concl$" при чему су asm_1, ..., asm_n претпоставке, *concl* је закључак, а x_1, ..., x_k су универзално квантификоване променљиве. Уместо **lemma** могуће је користити кључну реч **theorem** и нема никакве разлике јер је то у систему потпуно исто.

Ми смо у овом раду интензивно користили језик *Isar* који је део *Isabelle/HOL* система и чија архитектура омогућава писање структурираних доказа (припада групи декларативних језика). То заправо значи да је са језиком Isar могуће писати „читљиве" доказе. *Isar* је веома богат језик, и ми ћемо овде описати само оне конструкције које су коришћене у овом раду. Тривијални докази се показују позивом неког аутоматског метода, тј. тактике, навођењем иза кључне речи **by**. Сложенији докази писани у *Isar* стилу се задају између кључних речи **proof** и **qed**. У самом доказу могу постојати фиксне променљиве (задате кључном речи **fix**, претпоставке (**assume**), помоћна тврђења (**have**) и на крају и сам закључак теореме задат са кључном речи **show**. Са кључном речи **using** или са кључном речи **from** се уводе чињенице које су потребне да би се тврђење доказало. Ове чињенице могу бити претпоставке, помоћна тврђења или друга тврђења која су раније већ доказана. На пример:

```
lemma num_property:
    "∀ (x::real). (x − 1) · (x + 1) > 0 ⟶ x² > 1"
 proof-
    fix x::real
    assume "(x − 1) · (x + 1) > 0"
    have "(x − 1) · (x + 1) = x² − 1"
      by (simp add: power2_eq_square field_simps)
    then show "x² > 1"
      using `(x − 1) · (x + 1) > 0`
      by simp
qed
```

Методи за доказивање

До сада смо више пута спомињали методе за доказивање, тактике, симплификацију итд, али нисмо улазили у детаље. У овом поглављу ћемо укратко објаснити могућности за доказивање у систему *Isabelle/HOL* и објаснићемо њихову теоријску основу.

Природна дедукција

Основни метод за доказивање је природна дедукција о којој је било речи у поглављу 2.1. *Isabelle* користи одговарајућу колекцију правила природне дедукције да би доказ тражио аутоматски. Већ смо видели да за сваки везник, постоје правила која га уводе и правила која га елиминишу. Систем *Isabelle* има основу у правилима природне дедукције, па многи алати користе терминологију увођења или елиминисања правила.

Посматрајмо уопштено правило:

$$\mathtt{R} = \frac{P_1 \;\; P_2 \ldots \;\; P_n}{Q}$$

Методе које можемо применити у доказивању коришћењем овог правила су:

- `rule R` метода унификује Q са тренутним циљем и уводи n нових подциљева за доказивање инстанци $P_1, P_2, \ldots, P_n$. Заправо, ово правило у математичким текстовима (за везник $\wedge$) је

$$\frac{A \qquad B}{A \wedge B}$$

- `erule R` метода унификује Q са тренутним циљем, унификује P_1 са неком претпоставком и уводи $n-1$ нови подциљ за доказивање инстанци $P_2, \ldots, P_n$.

$$\frac{A \vee B \qquad \begin{array}{c}[A] \\ \vdots \\ C\end{array} \qquad \begin{array}{c}[B] \\ \vdots \\ C\end{array}}{\mathrm{C}}$$

- `drule R` метод унификује P_1 са неком претпоставком и одмах га брише и уводи n нових подциљева $P_2, \ldots, P_n$ и иницијални циљ, али са додатом

претпоставком Q. На пример, у математичким текстовима ово правило за везник $\wedge$ је

$$\frac{A \wedge B}{A}$$

- `frule R` метод се понаша као `drule R` метод, али не брише претпоставку коју је унификовао.
- `rule_tac` $v_1 = t_1$ **`and`** ... **`and`** $v_m = t_m$ **`in`** `R` метод је исти као и метод `rule R`, али иницијализује промењиве као што је наведено у правилу.

Правила за увођење служе да одреде када је могуће извести формулу која садржи одговарајући везник.

`conjI:` $[\![?P; ?Q]\!] \Longrightarrow ?P \wedge ?Q$
`disjI1:` $?P \Longrightarrow ?P \vee ?Q$
`disjI2:` $?Q \Longrightarrow ?P \vee ?Q$
`disjCI:` $(\neg ?Q \Longrightarrow ?P) \Longrightarrow ?P \vee ?Q$
`impI:` $(?P \Longrightarrow ?Q) \Longrightarrow ?P \longrightarrow ?Q$
`notI:` $(?P \Longrightarrow$ `False` $) \Longrightarrow \neg ?P$
`clasical:` $(\neg ?P \Longrightarrow ?P) \Longrightarrow ?P$
`allI:` $(\bigwedge x.?Px) \Longrightarrow \forall x.?Px$
`exI:` $?P?x \Longrightarrow \exists x.?Px$

Доказивање у систему *Isabelle* иде уназад — када применимо правило `conjI` подциљ је већ у облику конјукције и правило чини да везник $\wedge$ нестане. Можемо приметити да се у правилу користе шематске промењиве што омогућава да буду замењене било којом формулом. Задавањем правила у методи `rule` оно бива примењено на тренутно тврђење, што можемо видети у доњем примеру.

Правила за елиминацију служе да се ослободимо везника. Када примењујемо правила за елиминацију најбоље је користити метод `erule`. Правило `rule` најчешће производи један подциљ више него што то чини `erule`, а тај подциљ је углавном тривијалан.

`conjE:` $[\![?P \wedge ?Q;\ [\![?P; ?Q]\!] \Longrightarrow ?R]\!] \Longrightarrow ?R$
`disjE:` $[\![?P \vee ?Q; ?P \Longrightarrow ?R; ?Q \Longrightarrow ?R]\!] \Longrightarrow ?R$
`notE:` $[\![\neg ?P; P]\!] \Longrightarrow R$
`impE:` $[\![?P \longrightarrow ?Q; ?P; ?Q \Longrightarrow ?R]\!] \Longrightarrow ?R$

`allE:` $[\![\forall x.?Px; ?P?x \Longrightarrow ?R]\!] \Longrightarrow ?R$
`exE:` $[\![\exists x.?Px; \bigwedge x.?Px \Longrightarrow ?Q]\!] \Longrightarrow ?Q$
`contrapos_pp:` $[\![?Q; \neg ?P \Longrightarrow \neg ?Q]\!] \Longrightarrow ?P$
`contrapos_pn:` $[\![?Q; ?P \Longrightarrow \neg ?Q]\!] \Longrightarrow \neg ?P$
`contrapos_np:` $[\![\neg ?Q; \neg ?P \Longrightarrow ?Q]\!] \Longrightarrow ?P$
`contrapos_nn:` $[\![\neg ?Q; ?P \Longrightarrow ?Q]\!] \Longrightarrow \neg ?P$

Применом правила за деструкцију елиминише се неки везник. За њихово коришћење најбоље је користити метод `drule`.

`conjunct1:` $?P \wedge ?Q \Longrightarrow ?P$
`conjunct2:` $?P \wedge ?Q \Longrightarrow ?Q$
`mp:` $[\![?P \longrightarrow ?Q; ?P]\!] \Longrightarrow ?Q$

Посматраћемо пример који смо имали раније (Пирсов закон) и на коме можемо да видимо како примена правила утиче на процес доказивања.

```
lemma "((A ⟶ B) ⟶ A) ⟶ A"
apply (rule impI)
apply (rule classical)
apply (erule impE)
apply (rule impI)
apply (erule notE, assumption)+
done
```

Примена правила `impI` претвара циљ у нови циљ $(A \longrightarrow B) \longrightarrow A \Longrightarrow A$ што правилом `classical` постаје циљ $[\![(A \longrightarrow B) \longrightarrow A; \neg A]\!] \Longrightarrow A$. Применом методе `erule impE` уводе се два нова подциља $\neg A \Longrightarrow A \longrightarrow B$ и $[\![\neg A; A]\!] \Longrightarrow A$. Да би се ослободили везника $\longrightarrow$ у првом подциљу примењујемо правило `rule impI` и први подциљ постаје $[\![\neg A; A]\!] \Longrightarrow B$. Коначно, примењујемо `erule notE` на први подциљ и добијамо $A \Longrightarrow A$ што разрешавамо са методом `assumption`. Слично је и за други подциљ и са `+` смо означили да се иста правила и методе примене на други подциљ. Овим је доказ готов.

Isabelle такође користи и супституцију у доказивању. Правило супституције дозвољава да се терм s замени термом t ако можемо да докажемо $t = s$.

`ssubst:` $[\![?t = ?s; ?P?s]\!] \Longrightarrow ?P?t$

Симплификација

Једно од главних аутоматских метода које се примењује је *преписивање термова*, односно узастопна примена једнакости које важе. Наравно, треба бити пажљив у коришћењу преписивања јер може да траје бесконачно. Симплификација се покреће са `simp`. Метод `simp` се може задати и са *листом модификатора*, односно са `add` и `del` који служе да додају, односно да избришу нека правила која користи метод `simp`. На пример, **`apply`** `(simp add: is_empty_def)` додаје дефиницију у правила за упрошћавање (понекад се користи и **`unfolding`** `is_empty_def` уколико желимо само да развијемо дефиницију, а метод `simp` уради и много више од тога).

Аутоматски методи `blast` и `auto`

Често је приликом доказивања потребно пронаћи одређени редослед за примену правила да би доказ могао успешно да се изведе и да се заустављa. Поред тога, докази могу бити јако дуги ако се примењује у сваком кораку једно по једно правило. Да би убрзали претрагу правила, као и редослед којим се она примењују примењујемо аутоматске методе, као што је метод `blast`. Овим методом се комбинују сва горе наведена правила (правила за елиминацију, за деструкцију, правила за увођење, односно `conjI`, ..., `conjE`, ..., `contrapos_pp`, ..., `mp`, `ssubst`) док се не нађе неки доказ. Рецимо, Пирсов закон који смо имали раније и који смо доказивали у више корака се може у року неколико милисекунди доказати методом `blast`. Довољно је било да смо написали:

`lemma` "$((A \longrightarrow B) \longrightarrow A) \longrightarrow A$"
`by` `blast`

Метод `blast` такође може да ефикасано доказује многе теореме теорије скупова. Наиме, овај метод се заснива на методи таблоа која је имплементирана у језику *ML*. Токође, метод се ослања на велику базу лема коју може да претражује и која омогућава резоновање о скуповима (у оквиру система *Isabelle/HOL* развијена је Цермело–Френкел теорија скупова са аксиомом избора), функцијама и релацијама.

Метод који комбинује класично резоновање са упрошћавањем је метод `auto`. Његов циљ је да докаже једноставне подциљеве и делове подциљева.

Нажалост, може да произведе велики број подциљева јер док неке подциљеве доказује други дели и тако производи нове подциљеве.

Поред ових метода, постоје и бројни други, рецимо метод `force`, `clarify`, `smt` који покреће *SMT* доказивач [13] итд. Такође, постоје још и бројна правила која нисмо навели, на пример, `someI, someI2, order_antisym` итд. Овде нећемо улазити у све детаље јер то није циљ овог рада, а више се може погледати у [131].

Дефинисање функција

Функције у систему emphIsabelle/HOL се могу задати коришћењем примитивне или генералне рекурзије. Дефиниције функција користе функционалност уграђеног пакета за функције [96] и ми ћемо овде представити неке основне могућности. Функција се задаје својим именом, типом и скупом дефинисаних рекурзивних једначина. Све променљиве леве стране једначине морају бити различите.

```
fun ins :: "'a ⇒ 'a list ⇒ 'a list"
where
     "ins a (x#y#xs) = x # a # ins a (y#xs)"
   | "ins a [v] = [v, a]"
   | "ins a [] = []"
```

HOL је логика тоталних функција и заустављање функција је важан услов који спречава неконзистентност. На пример, из дефиниције `f(`n`) = f(`n`) +` 1 може се доказати да је $0 = 1$ ако би скратили `f(`n`)` са обе стране једнакости. Функција `ins` се увек зауставља јер њени аргументи постају мањи са сваким рекурзивним позивом. Када користимо **`fun`**, *Isabelle* покушава да докаже заустављање аутоматски. Али, понекад, у томе не успе и дефиниција бива одбијена. У том случају, обично је најбоља идеја користити шири облик дефиниције са кључном речи **`function`**. Коришћењем овог ширег облика корисник може да види где је настао проблем у доказивању. Синтакса је:

```
function fun_name :: T1 ⟹ T2 ⟹ ... where equations
⋮
by pat_completeness auto
termination by lexicographic_order
```

Дефиниција функције ствара неопходан услов којим се изражава комплетност и компатибилност и то се доказује (у овом примеру) коришћењем метода `pat_completeness` и `auto`. Доказ заустављања почиње после дефиниције, након команде **`termination`**. Метод `lexicographic_order` је подразумевани метод. Ако овај метод не успе, доказ се изводи ручно. У логици вишег реда све функције су тоталне и то је управо разлог зашто мора бити доказано заустављање јер то даје оправдање да је задатом рекурзијом функција добро дефинисана.

Када се докаже заустављање, систем *Isabelle/HOL* обезбеђује прилагођено индуктивно правило за сваку дефинисану функцију. Име тог правила је `fun_name.induct`, а доказ некада можемо започети именом методе које користимо за доказивање, на пример:
`proof` `(induct` v_1 $v_2 \ldots$ `rule: fun_name.induct)`
при чему су v_1, $v_2 \ldots$ променљиве које су задате у тврђењу (**`theorem, lemma, have,`** $\ldots$) као параметри `fun_name`.

```
lemma "ins a x @ ins a y = ins a (x @ y)"
proof (induct x rule:ins.induct)
   case (1 a x p q)
   assume"ins a (p # q) @ ins a y = ins a ((p # q) @ y)"
   thus ?case
     by simp
next
   case (2 a)
   show ?case
     by simp
next
   case (3 a v)
   show ?case
     apply (induct y rule:ins.induct)
     apply simp+
qed
```

Након извршења корака **`proof`** `(induct` x `rule:ins.induct)`
доказ се дели на три дела (подциља):

1. $\bigwedge a\ x\ ya\ xs.$

```
      ins a (ya # xs) @ ins a y = ins a ((ya # xs) @ y) ⟹
      ins a (x # ya # xs) @ ins a y = ins a ((x # ya # xs) @ y)
2. ⋀ a. ins a [] @ ins a y = ins a ([] @ y)
3. ⋀ a v. ins a [v] @ ins a y = ins a ([v] @ y)
```

Са сваким **case** кораком се доказује један од ових подциљева. Први подциљ је индуктивни корак, док су други и трећи подциљ база индукције. Можемо приметити да је за доказ трећег подциља коришћена индукција по другој листи (листи y), али како су кораци прилично једноставни није било потребе развијати доказ већ само позвати симплификатор.

Понекад је потребно дефинисати парцијалну функцију. Ово није могуће у HOL, али могуће је наместити да се тотална функција понаша као парцијална. Поред функције, чије је име `fun_name` дефинише се и `fun_name_dom` и он даје домен функције, односно вредности у којима функција сигурно зауставља. Индукциона правила домена омогућавају да се покаже да дата вредност лежи у домену функције ако сви аргументи у свим рекурзивним позивима такође леже у домену. Приликом доказивања индукције за такву функцију позива се правило `fun_name.pinduct` и при томе се индукција врши само за оне вредности које су дефинисане у домену, коришћењем `fun_name_dom`. Више о овоме може се наћи у [96].

Систем модула

Систем модула (енг. *locales*) [9] су механизам за структурну спецификацију и за модуларност у систему *Isabelle*. Са модулима се уводи слој апстракције тако што се фиксира скуп параметара (функција и константи) и потом се задају претпоставке о овим параметрима. Резоновање се ради апстрактно узимајући у обзир дате претпоставке, а резултати се могу користити за сваку конкретну константу или функцију која задовољава претпоставке дате у модулу (кажемо да оне интерпретирају модул). Синтакса за дефинисање модула је:

locale name

— Са овим задајемо име модула.

fixes param_name :: $T_1 \Longrightarrow T_2 \Longrightarrow \ldots$

— Параметар са синтаксом при чему су фиксирани параметри различити. Типови за параметре не морају бити иницијализовани. Ако ништа није

назначено, узимајући у обзир претпоставке, елементи могу имати и општи тип.

assumes ....
— Претпоставке.

begin
body
end

Имена у модулима су квалификована именом модула у коме су дефинисана, на пример `H.f` означава функцију f из теорије H. У оквиру контекста (између **body** и **end**) могу бити задате дефиниције и теореме. Пример за модуо којим се описује функција сортирања елемената листе је:

```
locale Sort =
  fixes sort :: "'a::linorder list ⇒ 'a list"
  assumes sorted: "sorted (sort l)"
  assumes permutation: "sort l <  > l"
```

Овим се дефинише да функција сортирања као улазни аргумент узима листу елемената који се могу поредити и враћа листу. При томе враћена листа мора бити сортирана (`sorted`) и заправо представљати пермутацију полазне листе (`permutation`). Ознаком `'a::linorder` је ограничено да тип `'a` припада класи `linorder`, тј. над том типу постоји линеарно уређење (више о класама видети у [71]).

Теореме и дефиниције задате у оквиру модела се могу користити и касније, у другим контекстима када важе претпоставке модула (као дефиниције или већ доказане теореме). Ово поновно коришћење теорема се зове интерпретација модула и постоји два начина како се то може учини, коришћењем команде **sublocale** или **interpretation**.

Команда **sublocale** омогућава да се идентификује веза између модула и зависних модула. Декларација **sublocale** $L_1 \geq L_2$ означава да се што је доказано у L_2 важи и у L_1. Додатно, дефиниција је динамична – сваки нови закључак који се касније дода у L_2 ће исто да важи и у L_1. Ова релација је и транзитивна.

Команда **interpretation** означава интерпретацију модула у некој теорији. Ово је слично инстанцирању конкретног објекта неког објектно оријентисаног

типа. Рецимо, хип сортирање би могло бити једна од интерпретација дефинисаног модула `Sort`.

Модули се често користе у верификацији софтвера јер омогућавају механизам који се назива „упрошћавање програма" [70].

Ми ћемо на више места користити модуле. Негде ћемо дефинисати своје модуле, а на неким местима ћемо користити модуле из постојећих *Isabelle/HOL* библиотека.

Количнички пакет

Други начин да се уведу нови типови, често коришћен и у математици, јесу количнички типови. Количнички типови су згодни за апстракцију типа: уместо да експлицитно доказујемо да функција задовољава релацију еквиваленције или да чува инваријанте, ова информација може бити енкодирана у самом типу функције. Као што смо раније видели, *Isabelle* захтева рад са функцијама `Rep` и `Abs` које служе као веза између старог типа и новог апстрактног типа. Зато дефинисање функција над апстрактним типом може бити незгодно и често захтева доказивање бројних тврђења. Зато постоје разни механизми који аутоматизују рад са количничким типом.

У *Isabelle/HOL* постоји више пакета који омогућавају рад са количничким типом, и у нашој формализацији ми смо користили *lifting/transfer* пакет [84].

Први корак у дефинисању количничког типа је дефиниција релације еквиваленције $\approx$ над неким постојећим (репрезентативним) типом τ. Количнички тип κ се онда дефинише са **`quotient_type`** κ `=` τ `/` $\approx$.

Прво имамо неки основни тип `'a` и релацију еквиваленције `R :: 'a ⇒ 'a ⇒ bool`. Потом имамо апстрактни тип `'b` који је у кореспоненцији један–на–један са класама еквиваленције релације `R`. Наиме, функција апстракције `Abs :: 'a ⇒ 'b` мапира сваку класу еквиваленције релације `R` у тачно једну апстрактну вредност, а функција репрезентације `Rep :: 'b ⇒ 'a` претвара апстрактну вредност у произвољан елемент одговарајуће класе еквиваленције. За дати тип `'a` и релацију `R`, команда **`quotient_type`** уводи нови тип и одговарајуће функције `Abs` и `Rep`.

Као пример узећемо релацију једнакости по модулу природних бројева. Прво дефинишемо релацију.

```
definition moduo :: "nat ⇒ nat ⇒ bool" (infix "≈" 50) where
```

```
"a ≈ b ⟷ (∃ m. m ≠ (0::nat) ∧ a mod m = b mod m)"
```

Да бисмо могли да дефинишемо количнички тип над овом релацијом потребно је доказати да је ова релација заправо релација еквиваленције. Зато доказујемо следеће леме:

```
lemma moduo_eq_refl: "a ≈ a"
lemma moduo_eq_sym:
   assumes "a ≈ b"
   shows "b ≈ a"
lemma moduo_eq_trans:
   assumes "a ≈ b" "b ≈ c"
   shows "a ≈ c"
```

Ове леме се тривијално доказују и у те детаље нећемо сада улазити. Потом дефинишемо количнички тип:

```
quotient_type
   moduo_qt = nat / "moduo"
```

Ова дефиниција изазива обавезу да се докаже да је коришћена релација релације еквиваленције. Добијамо подциљ облика: `equivp op ≈`, при чему `equivp` означава да треба показати особину еквиваленције.

И то се тривијално доказује коришћењем показаних лема на следећи начин:

```
apply (rule equivpI, rule reflpI, simp add: moduo_eq_refl)
apply (rule sympI, simp add: moduo_eq_sym)
apply (rule transpI, blast intro: moduo_eq_trans)
done
```

Функције над количничким типом се дефинишу у два корака. Прво, функција f_τ :: ... τ ... се дефинише над репрезентативним типом τ. Потом се користи *lifting* пакет да се функција „подигне" из базичног типа на апстрактан тип. Функција се подиже на количнички тип коришћењем **lift_definition** f_κ :: ... κ ... **is** f_τ. Ово ствара обавезу да се докаже да дефиниција не зависи од избора представника. Цео овај поступак посматрамо на једноставном примеру над апстрактним типом модула. Дефинисана функција над базичним типом над три аргумента је:

```
definition add_moduo :: "nat ⇒ nat ⇒ nat ⇒ nat" where
  "add_moduo a b c = (a + b) * c"
```

Потом подижемо ову дефининицију на ниво количничког типа:

```
lift_definition add :: "moduo_qt ⇒ moduo_qt ⇒ moduo_qt ⇒ moduo_qt" is
  add_moduo
```

и ова дефиниција ствара обавезу да се докаже:

$$\bigwedge x\ y\ z\ x_1\ y_1\ z_1.\ [\![\ x \approx x_1;\ y \approx y_1;\ z \approx z_1\ ]\!] \Longrightarrow \texttt{add\_moduo}\ x\ y\ z \approx \texttt{add\_moduo}\ x_1\ y_1\ z_1$$

што заправо представља тврђење да како год изабрали представнике класе резултат ће бити елемент тачно једне класе. Како се може тривијално доказати да плус и пута чувају модуо, доказ је прилично једноставан и захтева само додавање одговарајућих дефиниција типа и функције у скуп правила:

```
apply (simp add: add_moduo_def moduo_def)
by auto
```

Пакет *transfer* обезбеђује методу за доказивање `transfer` која замењује тренутни подциљ са логички еквивалентним који узима друге типове и константе. Односно, за количнички тип, метод `transfer` редукује тренутни подциљ који је везан за количнички тип на подциљ који је везан за основни, базични тип. Овај метод није строго везан само за количнички тип, може бити примењен и за друге типове ако постоје одговарајућа правила за трансфер. Када дефинишемо функцију коришћењем **`lift_definition`** аутоматски се генеришу правила за трансфер за дату функцију (током рада овај поступак се не примећује јер се одвија у позадини).

Узмимо да желимо да докажемо једноставно тврђење са нашу дефинисану функцију `add`: **`lemma`** "`add` $a\ b\ c$ = `add` $b\ a\ c$"

Када применимо **`apply transfer`** генерише се нови подциљ који је над представницима класа:

$\bigwedge a\ b\ c.$ `add_moduo` $a\ b\ c \approx$ `add_moduo` $b\ a\ c$

Обратимо пажњу да је релација једнакости замењена раније дефинисаном релацијом еквиваленције, односно како год изабрали представнике класе, резултат ће бити елемент једне те исте класе. Као и раније овај подциљ се врло лако доказује:

```
apply (simp add: add_moduo_def moduo_def)
by auto
```

Више детаља о количничком пакету се може пронаћи у литератури [91, 84]. У овом тексту користићемо неагресивну нотацију ($\lfloor\_\rfloor$ и $\lceil\_\rceil$) за функцију репрезентације и за функцију апстракције и игноришући ове ознаке текст може бити разумљивији и сличнији уобичајеним математичким текстовима.

Глава 3

Преглед формализације и аутоматског доказивања у геометрији

У последњих десет година направљени су значајни резултати у формализацији математике коришћењем интерактивних доказивача. У овом поглављу навешћемо неке најзначајније радове (како класичне, тако и најновије) у формализацији геометрије. У сваком поглављу радови су наведени хронолошки према годинама када су објављени.

3.1 Аутоматско доказивање у геометрији

У раду „Аутоматско резоновање у геометрији" [29] постоји веома детаљан историјски опис развоја аутоматских доказивача до почетка овог века, као и детаљан опис различитих приступа у аутоматском доказивању у геометрији и нешто од тих идеја је приказано у овом поглављу.

Алгебарски доказивачи

Вуов метод и метод Гребнерових база. Највећи напредак у аутоматском доказивању теорема у геометрији направио је Ву (енг. *Wu*). Он је ограничио скуп проблема које је разматрао на проблеме са једнакостима. На такве проблеме је могао да примени моћан метод који је могао да докаже и компликована тврђења. Овај метод је представио у оквиру свог рада [170] 1978. годи-

не. Како је овим методом показано 130 тврђења (међу којима је Фојербахова (нем. *Feuerbach*) теорема, Карнотова (фр. *Carnot*) теорема, тангента–секанс теорема, Птоломејева теорема, Ојлерова теорема, Стјуартова теорема и многе друге) [27], он постаје све популарнији. Бројни аутори су овај метод имплементирали и унапређивали разним хеуристикама [51, 150, 92]. Убрзо, постаје јасно да се Вуов приступ могао извести из Ритовог (енг. *Ritt*) рада [146], па се често овај метод још назива и *Ву–Рит метод*.

Успех овог метода утицао је на развој нових метода. Један од успешних је метод Гребнерових база, који се заснива на *Бухбергеровом алгоритму* (нем. *Buchberger*) [24] и може се применити на исту класу проблема као и Вуов метод. Бухбергеров алгоритам су унапређивали бројни аутори и данас постоје разне хеуристике које служе да повећају ефикасност алгоритма. Коришћењем овог метода показана су многа геометријска тврђења [25], као што су Гусова теорема, Папусова теорема, Дезаргова теорема, теорема о Ојлеровој правој троугла и многе друге. Постоје бројне имплементације, а неке и у комерцијалним програмима (нпр. *Matlab* и *Mathematica*).

Главна мана ова два метода је што се са њима *не могу доказивати неједнакости* и самим тим се не могу разматрати теореме које говоре о распореду тачака. За решавање проблема са неједнакостима, Ву је предложио метод који се заснива на проналажењу минималне или максималне вредности полиномијалне функције под одређеним условима [165]. Поред доказивања у елементарној геометрији, Ву је представио и *метод за доказивање у диференцијалној геометрији* [164]. Постоје и проширења која омогућавају да се *метод користи и за хиперболичку геометрију* [172].

Полусинтетички доказивачи

Метод површина и његова проширења. Сви набројани методи преводе геометријско тврђење у једначине коришћењем координата тачака које се посматрају, а потом се примењују алгебарске технике на ове једначине. Ови доказивачи дају одговор „да" или „не", али не дају никакву информацију о извођењу која би била разумљива човеку и слична доказима у школским уџбеницима. Постоје бројни покушаји да се направе доказивачи који би поред доказивања уједно производили *читљиве доказе*. Један од најзначајнијих је *метод површина* [30]. Овај метод користи геометријске величине као што су површина, размера, Питагорина разлика и слично. Главна предност овог

доказивача је што сваки корак у доказивању има јасно геометријско значење. Додатно, експерименти су показали да су докази коришћењем метода површина краћи. Метод је могуће проширити тако да је могуће радити и са неједнакостима. Главна идеја метода је изразити хипотезе теореме коришћењем скупа конструктивних тврђења, од којих свако тврђење уводи нову тачку или нову праву и представља закључак као једнакост израза у којем се налазе геометријске величине, а без коришћења Декартових координата. Доказ се заснива на елиминацији (у обрнутом редоследу) тачака и линија коришћењем одговарајућих лема. Када се елиминишу сви уведени елементи, тренутни циљ постаје једнакост између два израза са геометријским величинама датих само слободним тачкама. Ако је ова једнакост тривијално тачна, онда је оригинално тврђење доказано; ако је тривијално нетачна, онда је доказано да је почетна претпоставка нетачна; иначе, тврђење није ни доказано, ни оповргнуто.

Чу (енг. *Chou*) и сарадници су такође представили и *метод пуног угла* [33]. Метод пуног угла као величину користи новоуведени појам *пун угао* преко ког изражава односе између тачака и дужи. *Пун угао* је дат као уређен пар правих, а веома је битан однос између два пуна угла, тј. угао одређен правама m и n је једнак пуном углу одређеном правама u и v ако постоји ротација ρ таква да $\rho(m)||u$ и $\rho(n)||v$. У експериментима је примећено да се метод површина веома добро понаша за конструктивне теореме у афиној геометрији. Са друге стране, метод пуног угла је погодан за проблеме у којима има много кругова и углова и за овакве проблеме чешће производи краће доказе него што је то случај са методом површина. Овај метод се користи као допуна метода површина.

Исти аутори су у раду [32] представили могућност *проширења методе површина на проблеме у стереометрији*. Хипотезе се задају конструктивно, а закључци су полиномијалне једначине неколико геометријских величина, као што су запремина, размера дужи, размера површина и Питагорине разлике. Главна идеја овог метода је да елиминише тачке из закључка геометријског тврђења коришћењем основних својстава запремине.

У раду из 2016. године, Шао (енг. *Shao*), Ли (енг. *Li*) и Хуанг (енг. *Huang*) [152] посматрају задатке из стереометрије са Олимпијских такмичења из математике. Они у раду представљају три различита проблема и дају полиноме које су извели на папиру и којима се у полиномијалном облику задају

посматрана геометријска тврђења. Коришћењем ова три примера они показују да се алгебарски методи могу користити за доказивање у стереометрији. За сваки пример користили су три различита метода: метод карактеристичног скупа [171, 162, 50, 26], метод Гребнерових база [41, 98, 156, 35] и метод вектора [106]. Методи се пореде и закључак је да метод вектора даје бољи геометријски доказ, али формуле могу бити дуге и незгодне за манипулацију и израчунавање. Ипак, они не нуде неки систематичан начин како се геометријска тврђења могу аутоматски алгебризовати, односно представити полиномима.

Поред ова два поменута рада, није нам познато да постоји још радова који се баве аутоматским доказивањем у стереометрији.

Иако је метод површина описан још деведесетих година прошлог века, до скора нису детаљно описана имплементациона питања, али ни испитана оправданост коришћења самог метода. У раду [87] аутори управо скрећу пажњу на ове проблеме. Они *веома детаљно описују метод и формално доказују у систему Coq важне дефиниције и леме које омогућавају коришћење метода*. Детаљно описују и нека важна имплементациона питања, јер метод површина, иако је једноставан за разумевање, је тежак за имплементацију јер постоји много детаља на које би требло обратити пажњу.

Синтетички доказивачи

Стојановић са сарадницима демонстрира коришћење *синтетичког доказивача за доказивање теорема у геометрији Тарског* [159]. Пре свега је интересантно то што је повезано интерактивно и аутоматско доказивање, што значи да су сви аутоматски генерисани докази уједно и формално верификовани. Систем поред доказивања теорема генерише и машински проверене, читљиве доказе који су веома слични доказима из уџбеника. Аутори користе кохерентну логику, део логике првог реда као основну логику система. Примењују резолуцијски доказивач, доказивач теорема у кохерентној логици, и *XML* алатке за кохерентну логику који им омогућавају да доказе трансформишу у машински проверљиве доказе и у доказе разумљиве човеку. Систем примењују на доказивање тврђења из првог дела књиге „Математички методи у геометрији" [147] и успешно, потпуно аутоматски доказују 37% теорема.

Сличан приступ је описан у раду [12] у којем се описује *полуаутоматски приступ за доказивање у геометрији Тарског*. Аутори су посматрали више

група доказа, од којих су неки краћи од 40 корака, потом доказе који су између 40 и 100 корака и који се сматрају тежим за човека и коначно доказе дуже од 100 корака који најчешће представљају теме докторских радова. За доказивање користе доказивач *OTTER*, који је и раније коришћен за доказивање теорема [142]. Један од циљева је била и анализа утицаја хардверског напретка, али и нових техника за аутоматско доказивање у геометрији на успешност аутоматских доказивача. Потпуно механички је изведена већина кратких доказа. Испрва за дугачке доказе није било могуће добити механичке доказе, и аутори су коришћењем доказа из књиге конструисали формалне доказе. Потом су применили нову технику, где су доказивачу прослеђивали све потребне аксиоме и претходно доказана тврђења, као и неке кораке доказа и систем је успевао да нађе механичке доказе за тврђења која су се доказивала у више од 100 корака.

Остали доказивачи и примене

Један од првих радова у области аутоматског доказивања у геометрији је Гелертнеров (енг. *Gelernter*) рад [52], који је користио методе вештачке интелигенције, и његов приступ се заснивао на прављењу доказа сличних онима које пише човек.

Новији рад који такође користи методе вештачке интелигенције је систем *Geometrix* [68].

Вос (енг. *Wos*) и његови сарадници [120] су користили резолуцијски доказивач за доказивање у геометрији Тарског. Програм је тестиран на проблемима из геометрије Тарског, али поред тога и на алгебарским проблемима, теорији категоричности и на проблемима верификације програма.

Додатно, постоји и рад који представља приступ за доказивање геометријских тврђења коришћењем елиминације квантора у линеарним и квадратним формулама над реалним бројевима [46]. Приступ се може користити за доказивање у реалној равни или вишим димензијама. За разлику од других приступа, овом методом се могу показати тврђења која иначе не могу да се покажу у пољу комплексних бројева. Додатно, формулација проблема може садржати и неједнакости (јер не морају теореме да буду универзално квантификоване) и алтернације квантификатора. Услови недегенерисаности се генеришу аутоматски, а чак је могуће аутоматски утврдити за тврђења која

не важе у уопштеном случају који услови недегенерисаности и које претпоставке су потребне да би тврђење важило.

Интензивно поље истраживања није само аутоматско доказивање него и коришћење аутоматских доказивача за решавање проблема у геометрији. У раду Весне Маринковић са сарадницима [114] аутори се баве *проблемом конструкције помоћу лењира и шестара*. У овом раду, фокус је на проблему конструкције троугла где су дате тачке и услови који за њих морају да важе. Приликом проналажења конструкције пролази се кроз све четири фазе (анализа, конструкција, доказ и дискусија), аутоматски се проналазе кораци конструкције, као и доказ исправности који се даје у облику који је читљив човеку. Додатно, користе се алгебарски методи за аутоматско доказивање да ли је могуће извршити конструкцију или није могуће извршити конструкцију, што је посебно интересантно јер постоји много проблема који су неконструктивни.

Различити системи за задавање конструкција и аутоматско доказивање у геометрији и њихова примена у образовању

Ванг (енг. *Wang*) је у раду [163] описао систем *GEOTHER* који може послужити за аутоматско доказивање теорема у геометрији. За развој система користи *Maple*. *Maple* омогућава симболичко и нумеричко израчунавње који садржи динамички типизиран императивни програмски језик који највише личи на *Pascal*. Поред тога има и могућност визуелизације, анализе података и рада са матрицама. Коришћењем програма *Maple*, Ванг геометријска тврђења репрезентује коришћењем предикатске спецификације, а те спецификације је могуће аутоматски превести на тврђења записана на енглеском или кинеском, или на алгебарске једнакости или на логичке формуле. На основу спецификација могуће је конструисати и дијаграме које је потом могуће мењати коришћењем миша. Оно што је посебно интересантно је што је у систему имплементирано више аутоматских доказивача теорема. Имплеметиран је доказивач заснован на Вуовој методи. Потом доказивач заснован на Кацлер—Стифтер (нем. *Kutzler–Stifter*) методи и доказивач заснован на Капуровој (енг. *Kapur*) методи (оба метода су заснована на идејама методе Гребнерових база). Имплеметирани су још и методи засновани на нула

декомпозицији и обичној диференцијалној нула декомпозицији [161]. Доказивачи су упоређивани над више различитих теорема, а Вуов метод се за већину тврђења показао као најефикаснији.

Значајно је поменути и систем *Geometry Expert* [34] који има имплементиран Вуов метод, метод Гребнерових база, метод вектора, метод пуног угла и метод површина. Посебно је интересантно његово проширење, систем *Java Geometry Expert* [173]. Ова алатка је занимљива јер поред аутоматског доказивања теорема нуди и визуелну, динамичку репрезентацију доказа. Производи серију визуелних ефеката за презентацију доказа и у својој бази садржи преко шест стотина примера.

Систем *Geometry Explorer* [168] производи читљиве доказе о својствима конструисаних објеката коришћењем метода пуног угла.

Важно је поменути и систем *Discover* [17] за аутоматско откривање у Еуклидској геометрији, који користи алгебарски софтвер *CoCoA* [1] и *Mathematica* [169].

Значајан је и систем *GCLC* [86] који омогућава запис конструкције и тврђења и превођење истих у различите формате (рецимо, у формат .tkz који је значајан јер је погодан за уметање слика у *TeX* документ). Овај систем је посебно значајан јер има интегрисана три доказивача теорема: Вуов метод, метод Гребнерових база и метод површина.

Нешто другачији приступ у односу на поменуте системе има систем *Cinderella* [95] који омогућава интерактивно задавање конструкције, а онда проверава да ли је добијена конструкција заиста тражено решење. Обављају се насумичне провере теорема да би се анализирале акције корисника, али се не даје доказ за дату конструкцију у било којој форми. Ово је систем који није симболички, ни динамички већ користи пробабилистичке методе да провери да ли је дата претпоставка највероватније теорема. Метод је заснован на леми Шварц–Зифел (нем. *Schwartz–Zippel*) која одређује број нула мултиваријантног полинома за дати максимални укупни степен. Више о свему овоме се може пронаћи у [145]. Оно што је важно је да програм може брзо да одлучи да ли два елемента у конструкцији су иста јер их теорема на то приморава или не. Програм враћа „није теорема" као одговор ако није успео да докаже теорему иако је могуће теорему доказати другим методама. Са друге стране, програм никада не враћа „јесте теорема" када то није случај. Систем *Cinderella* омогућава наивну подршку за нееуклидске геометрије. Софтвер је писан

у програмском језику *Java* па се може користити на више платформи.

Веома широко распрострањени у образовању су *динамички геометријски алати*. Коришћењем тог софтвера корисник лако може креирати и мењати геометријске конструкције. Конструкција обично започиње задавањем тачака или неких објеката (праве, кругови), а онда се креирају зависне тачке и објекти. Потом, почетна конфигурација се може мењати и може се посматрати како мењање почетних положаја утиче на крајњи резултат. На тај начин могуће је тестирати претпоставке, рецимо, да ли су три тачке увек колинеарне без обзира на конфигурацију. Како је могуће само тестирање, али не и доказивање, нови правци у развоју оваквог софтвера су управо у додавању аутоматских доказивача у оквиру динамичке геометријске алатке. Најпознатији динамичко геометријски софтвер је *GeoGebra* [1] и користи се у многим земљама, укључујући и Србију, као помоћно наставно средство.

Алгоритам који је проширење метода Гребнерових база и који се заснива на анализи система са параметрима се користи у раду [14] ради *проналажења хипотеза које морају да важе (поред претпоставки које су већ дате) да би дато тврђење било могуће извести*. Систем је имплементиран тако да се може користити у оквиру система *GeoGebra*. Значајно је што систем симболички одређује геометријско место тачака и потом показује валидност геометријског тврђења. Посебно је значајно и то што се у процесу одређивања геометријског места тачака, нотирају и ирелевантне тачке (најчешће су то у питању дегенерисани случајеви) и избацују из разматрања.

Поред поменутог, значајан *додатак систему GeoGebra је аутоматски доказивач заснован на Вуовој методи* [15]. Поред могућности доказивања, систем може да идентификује „интересантна" својства дате конструкције, односно да аутоматски одреди неке релације између конструисаних објеката.

3.2 Интерактивно доказивање у геометрији

Интерактивно доказивање у Хилберовој геометрији

Постоји велики број формализација фрагмената различитих геометрија у оквиру интерактивних доказивача теорема. Први покушај да се формализује *прва група Хилбертових аксиома и њихове последице* је био у оквиру

[1] `https://www.geogebra.org/`

асистената за доказивање теорема *Coq*, у интуиционистичком окружењу [45]. Следећи покушај је био у систему *Isabelle/Isar* и ову формализацију су радили Меикле (фр. *Meikle*) и Флорио (фр. *Fleuriot*) [121]. Аутори оповргавају уобичајено мишљење да су Хилбертови докази мање интуитивни, а више ригорозни. Важан закључак је да је Хилберт користио бројне претпоставке које у формализацији са рачунаром нису могле да буду направљене и стога су морале да буду формално верификоване и оправдане. Наставак ове формализације, пратећи Хилбертову књигу „Основи геометрије" [81], урадио је Скот у оквиру своје мастер тезе [149].

У раду [144] је *предложен минималан скуп Хилбертових аксиома* и за модел је коришћена теорија скупова. Изведена су и формално показана основна својства и тврђења у овом моделу.

Интерактивно доказивање у геометрији Тарског

Велике делове геометрије Тарског [147] је формализовао Нарбу (фра. *Narboux*) у систему *Coq* [127]. Бројна геометријска својства су изведена, доказано је више облика Пашове (нем. *Pacsh*) аксиоме, показана су бројна својства релације *подударно* и релације *између*. Рад се завршава доказом о постојању средишта тачке дужи.

У оквиру своје мастер тезе Макариос (енг. *Makarios*) је показао *независност аксиоме паралелности* [111]. За доказивање је изабрао аксиоматски систем Тарског зато што је тај систем категоричан. Да би могао да покаже независност, прво је формализовао аксиоме Тарског у оквиру система *Isabelle*. Онда је формализовао и Клајн Белтрами модел (енг. *Klein–Beltrami model*) нееуклидске геометрије Тарског и показао је да је ово модел за све аксиоме Тарског осим за аксиому паралелности, односно Еуклидову аксиому. На тај начин је показао да је ова аксиома независна од осталих аксиома Тарског за планарну геометрију. За неке аксиоме Тарског су у литератури недостајали докази да Клајн–Белтрами модел задовољава аксиому или су докази били некомплетни, па је рад попунио ове празнине. Као део рада, дефинисана је реална пројективна раван у систему *Isabelle/HOL* и показане су неке њене карактеристике.

Формализацију еквиваленције између различитих верзија *Еуклидовог петог постулата* дали су Бутри (фра. *Boutry*), Нарбу и Шрек (фра. *Schreck*) [19]. Овај постулат је посебно значајан јер је било много покушаја да се он до-

каже. Наиме, иако је пети постулат Еуклид записао као аксиому, веома рано је настала идеја да би он могао да се изведе из прва четири постулата. Бројни покушаји да се постулат докаже су били погрешни јер су се у доказима често користиле претпоставке које нису биле доказане. Аутори су у раду показали да је десет различитих тврђења еквивалентно Еуклидовом петом постулату. Такође, они су у раду разматрали како избор различитих верзија Еуклидовог постулата утиче на проблем одлучивања у геометријским доказима.

У раду [22] аутори су формализовали *првих дванаест поглавља књиге „Математички методи у геометрији"* [147] и на основу доказаних својстава су механички успели да докажу да се аксиоме Хилберта могу извести из аксиома Тарског. Браун (фра. *Braun*) и Нарбу су формализовали синтетички доказ Папусове теореме у геометрији Тарског [23]. Ова теорема је веома важна за *конструкцију координатне равни и представља један од важних корака у успостављању везе између аналитичке и синтетичке геометрије.* Ова веза је посебно важна јер омогућава коришћење алгебарског приступа у аутоматском закључивању у геометрији. Поред појмова који су дати у књизи, аутори су дали и формализацију вектора, четвороуглова, паралелограма, пројекције, оријентације праве и другог. Наставак овог рада и коначни производ вишегодишњег пројекта (први рад је објављен 2012. године) приказан је у раду [18] из 2016. године. Аутори су завршили формализацију књиге „Математички методи у геометрији" и у овом раду су показали како су формализвали последња три поглавља. Ову формализацију су искористили да *геометријски дефинишу аритметичке операције и доказали су да ове операције чине уређено поље. Потом су увели Декартову координатну раван и показали су својства основних геометријских релација. Ови резултати су веома важни јер оправдавају коришћење алгебарских метода за доказивање у геометрији.* Аутори то и демонстрирају у раду тако што користе метод Гребнерових база да докажу теорему о девет тачака на кругу.

Интерактивно доказивање у неколико различитих геометрија

Маго (фр. *Magaud*), Нарбу и Шрек су урадили још једну формализацију коришћењем система *Coq* и то за *геометрију пројективне равни* [108, 109]. Показана су нека основна својства и доказан је принцип дуалности за пројек-

тивну геометрију. Коначно, доказана је конзистенција аксиома у три модела, од којих су неки коначни, а неки бесконачни. На крају аутори дискутују о дегенерисаним случајевима и да би се са њима изборили користе рангове и монотоност.

Кан (енг. *Kahn*) је формализовао *вон Платову (фин.* von Plato*) конструктивну геометрију* такође у систему *Coq* [160, 90].

Потом, Гиљо (фра. *Guilhot*) користећи *Coq повезује Софтвер за интерактивну геометрију (СИГ) и формално доказивање* у намери да олакша учење еуклидске геометрије у средњој школи [69]. Фам (фра. *Pham*), Берто (фра. *Bertot*) и Нарбу су предложили и неколико унапређења [139]. Прво је да се елиминишу сувишне аксиоме коришћењем вектора. Они су додали четири аксиоме да опишу векторе и још три аксиоме да дефинишу еуклидску раван и увели су додатне дефиниције да би описали геометријске концепте. Коришћењем ових аксиома и дефиниција, геометријска својства су лако доказана. Друго унапређење је коришћење методе површина за аутоматско доказивање теорема. Да би се формално оправдало коришћење методе површина, морала је да се конструише Декартова координатна раван коришћењем геометријских својстава која су раније доказана.

Дупрат (фр. *Duprat*) формализује *геометрију лењира и шестара* [47]. Авигад (енг. *Avigad*) нуди још једну аксиоматизацију еуклидске геометрије [6]. Он полази од чињенице да еуклидска геометрија описује природније геометријска тврђења него новије аксиоматизације геометрије. Он сматра да посматрање слике, односно дијаграма, није пуно мана као што многи мисле. У намери да ово докаже, уводи систем E у коме су основни објекти тачке и праве. Аксиоме се користе да опишу својства дијаграма на основу којих се може закључивати. Аутори такође илуструју логички оквир у коме се могу изводити докази. У раду су презентовани неки докази геометријских својстава, као и доказ еквивалентности између система Тарског за геометрију лењира и шестара и система E. Дегенерисани случајеви су избегнути коришћењем претпоставки и стога се доказује само централни случај.

Као део пројекта *Flyspeck*, Харисон је развио веома богату теорију (која укључује алгебру, топологију и анализу) *Еуклидског n-димензионог простора* $\mathbb{R}^n$ у доказивачу теорема *HOL Light* [76, 80].

Показани су и различити резултати из *комплексне анализе* у оквиру доказивача теорема. Милевски (пол. *Milewski*) је доказао основну теорему алгебре

у систему Мизар [122], Хуверс (фр. *Geuvers*) са сарадницима је показао исту теорему у систему *Coq* [57], а Харисон је имплементирао комплексну елиминацију квантификтора за логику вишег реда и то је користио у разним формализацијама, укључујући и формализације геометрије.

3.3 Везе између интерактивних и аутоматских доказивача

Поред формализације геометријских тврђења многи истраживачи су покушали да формализују аутоматско доказивање у геометрији.

Грегоар (фр. *Grégoire*), Потје (фр. *Pottier*) и Тери (фр. *Théry*) комбинују модификовану верзију *Бухбергеровог алгоритма* и неке технике рефлексије да би добили ефективну процедуру која аутоматски производи формални доказ теорема у геометрији [67].

Женево (фр. *Génevaux*), Нарбу и Шрек су формализовали *упрошћен Вуов метод* у систему *Coq* [53]. Њихов приступ се базира на верификацији сертификата које генерише програм за упрошћен Вуов метод писан у *Ocaml*.

Фуч (фр. *Fuchs*) и Тери су формализовали Грасман—Кајл (енг. *Grassmann—Cayle*) алгебру у систему *Coq* [49]. Други део рада, који је интересантнији са аспекта наше формализације, представља *примену алгебре на геометрију инциденције*. Тачке, праве и њихови односи су дефинисани у форми алгебарских операција. Коришћењем ових дефиниција, Папусова теорема и Дезаргова теорема су интерактивно доказане у систему *Coq*. Коначно, аутори описују аутоматизацију у систему *Coq* за доказивање теорема у геометрији коришћењем ове алгебре. Мане овог приступа су у томе што је могуће показати само она тврђења где се доказује колинеарност међу тачкама и што се разматрају само недегенерисани случајеви.

Програме за *одређивање Гребнерове базе*, *F4* и *GB*, презентује Потје [140] и упоређује их са *gbcoq* [141]. Он предлаже решење са сертификатима и ово скраћује време које је потребно за израчунавања, тако да *gbcoq*, иако направљен у систему *Coq*, постаје упоредив са друга два програма. Примена Гребнерових база на алгебру, геометрију и аритметику је приказана кроз три примера.

Веома интересантан је рад који су представили Браун и Нарбу који се бави *проналажењем специјалних тачака троугла и доказивањем својстава које те*

тачке задовољавају [129]. Наиме, под руководством Кимберлинга (енг. *Kimberling*) направљена је електронска енциклопедија важних тачака троугла у којој се тренутно налазе дефиниције о више од 7000 тачака, као и својства које те тачке задовољавају. Ипак, ова својства често немају доказ или је доказ задат неформално. Како се у енциклопедији налази веома велики број тачака и њихових својстава, било би прилично напорно ручно преписивати и доказивати сва та својства. Браун и Нарбу су у оквиру система *Coq* дефинисали аутоматске методе за дефинисање тачака и аутоматске методе за доказивање њихових својстава. Веома важну улогу имају геометријске трансформације које помажу и у налажењу тачака и у доказивању одговарајућих лема.

Ботри (фр. *Boutry*), Нарбу и Шрек су у систему *Coq формализовали и имплементирали рефлексивну тактику за аутоматско генерисање доказа о инциденцији* [20]. Тврђења о инциденцији се често јављају у формалним доказима разних геометријских тврђења, али су у доказима који су записани на папиру често изостављена јер често не доприносе разумевању доказа. Ипак, приликом формалне верификације у оквиру асистента за доказивање теорема, леме и докази о инциденцији морају бити записани. Аутори су представили генеричку тактику која је примењива на било коју теорију чији је циљ да аутоматски докаже та ситна тврђења. Уједно, ово је један од низа корака да се формални доказ приближи доказима из уџбеника у којима се често изостављају „очигледна" тврђења.

Глава 4

Формализација аналитичке геометрије

4.1 Увод

Синтетичка геометрија се обично изучава ригорозно, као пример ригорозног аксиоматског извођења. Са друге стране, аналитичка геометрија се углавном изучава неформално. Често се ова два приступа представљају независно и веза између њих се ретко показује. Овај рад покушава да премости и више празнина за које мислимо да тренутно постоје у формализацији геометрије.

1. Прво, наш циљ је да формализујемо аналитичку геометрију, тј. Декартову раван у оквиру интерактивног доказивача теорема, са ригорозним приступом, али веома блиско стандардном средњошколском образовању. Представићемо добро изграђену формализацију Декартове геометрије равни у оквиру система *Isabelle/HOL*.

2. Намеравамо да докажемо да су различите дефиниције основних појмова аналитичке геометрије које можемо видети у литератури заправо еквивалентне, и да заправо представљају јединствен апстрактни ентитет – Декартову раван. Дефиниције ћемо преузети из стандардних уџбеника.

3. Намеравамо да докажемо да стандардна геометрија координатне равни представља модел аксиоматског система Тарског. Наиме, доказаћемо да Декартова координатна раван задовољава све аксиоме Тарског.

4. За већину аксиома Хилберта доказаћемо да их задовољава Декартова координатна раван.

5. Потом, намеравамо да анализирамо доказе и да упоредимо који од два система аксиома, систем аксиома Тарског или Хилбертов систем аксиома, је лакши за формализацију.

Поред тога што су многе теореме формализоване и доказане у оквиру система *Isabelle/HOL*, ми такође дискутујемо и наше искуство у примени различитих техника за поједностављење доказа. Најзначајнија техника је „без губитка на општости" ("бгно"), која прати приступ Харисона [78], а формална оправданост овог приступа је постигнута коришћењем различитих изометријских трансформација.

4.2 Формализација геометрије Декартове равни

Када се формализује теорија, мора се одлучити који појмови ће бити основни, а који појмови ће бити дефинисани помоћу тих основних појмова. Циљ наше формализације аналитичке геометрије је да успостави везу са синтетичком геометријом, па зато има исте основне појмове који су дати у синтетичком приступу. Свака геометрија има класу објеката који се називају *тачке*. Неке геометрије (на пример, Хилбертова) имају и додатни скуп објеката који се називају *праве*, док неке геометрије (на пример, геометрија Тарског) праве уопште не разматрају као примитивне објекте. У неким геометријама, праве су дефинисани појам, и оне су дефинисане као скуп тачака. Ово подразумева рад са теоријом скупова, а многе аксиоматизације желе то да избегну. У нашој формализацији аналитичке геометрије, ми ћемо дефинисати и тачке и праве јер желимо да омогућимо анализу и геометрије Тарског и геометрије Хилберта. Основна релација која спаја тачке и праве је релације *инциденције*, која неформално означава да права садржи тачку (или дуално да се тачка налази на правој). Други примитивни појмови (у већини аксиоматских система) су релација *између* (која дефинише редослед колинеарних тачака) и релација *подударности*.

Важно је напоменути да су у аналитичкој геометрији многи појмови често дати у облику дефиниција, а заправо ти појмови су изведени појмови у синте-

тичкој геометрији. На пример, у књигама за средњу школу дефинише се да су праве нормалне ако је производ њихових праваца -1. Ипак, ово нарушава везу са синтетичком геометријом (где је нормалност изведени појам) јер би оваква карактеризација требало да буде доказана као теорема, а не узета као дефиниција.

Тачке у аналитичкој геометрији.

Тачка у реалној координатној равни је одређена са својим x и y координатама. Зато, тачке су парови реалних бројева ($\mathbb{R}^2$), што се може лако формализовати у Isabelle/HOL систему са `type_synonym point`^{ag} $=$ "$real \times real$".

Редослед тачака.

Редослед (колинеарних) тачака се дефинише коришћењем релације *између*. Ово је релација која има три аргумента, $\mathcal{B}(A, B, C)$ означава да су тачке A, B, и C колинеаране и да је тачка B између тачака A и C. Ипак, неке аксиоматизације (на пример, аксиоматизација Тарског) дозвољава случај када је тачка B једнака тачки A или тачки C. Рећи ћемо да је таква релација*између инклузивна*, док неке друге аксиоматизације (на пример, Хилбертова аксиоматизација) не дозвољавају једнакост тачака и тада кажемо да је релација *између ексклузивна*. У првом случају, тачке A, B и C задовољавају релацију *између* ако постоји реалан број $0 \leq k \leq 1$ такав да $\overrightarrow{AB} = k \cdot \overrightarrow{AC}$. Желимо да избегнемо експлицитно коришћење вектора јер су они чешће изведени, а ређе примитиван појам у синтетичкој геометрији, тако да релацију *између* формализујемо у Isabelle/HOL систему на следећи начин:

```
definition "B^ag_T (xa, ya) (xb, yb) (xc, yc) ⟷
  (∃(k :: real). 0 ≤ k ∧ k ≤ 1 ∧
    (xb − xa) = k · (xc − xa) ∧ (yb − ya) = k · (yc − ya))"
```

Ако захтевамо да тачке A, B и C буду различите, онда мора да важи $0 < k < 1$, и релацију ћемо означавати са $\mathcal{B}^{ag}_H$.

Подударност.

Релација *подударно* дефинише се на паровима тачака. Неформално, $AB \cong_t CD$ означава да је дуж AB *подударана* дужи CD. Стандардна метрика у $\mathbb{R}^2$

дефинише да је растојање међу тачкама $A(x_A, y_A)$, $B(x_B, y_B)$ једнако $d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$. Квадратно растојање се дефинише као $d^2_{ag}\ A\ B = (x_B - x_A)^2 + (y_B - y_A)^2$. Тачке A и B су подударне тачкама C и D ако и само ако $d^2_{ag}\ A\ B = d^2_{ag}\ C\ D$. У *Isabelle/HOL* систему ово се може формализовати на следећи начин:

definition "$d^2_{ag}\ (x_1, y_1)\ (x_2, y_2) = (x_2 - x_1) \cdot (x_2 - x_1) + (y_2 - y_1) \cdot (y_2 - y_1)$"
definition "$A_1B_1 \cong^{ag} A_2B_2 \longleftrightarrow d^2_{ag}\ A_1\ B_1 = d^2_{ag}\ A_2\ B_2$"

Права и инциденција.

Једначина праве. Праве у Декартовој координатној равни се обично представљају једначинама облика $Ax + By + C = 0$, па тако тројка $(A, B, C) \in \mathbb{R}^3$ означава праву. Ипак, тројке у којима је $A = 0$ и $B = 0$ морају бити изузете јер не представљају исправну једначину праве. Такође, једначине $Ax+By+C = 0$ и $kAx+kBy+kC = 0$, за реално $k \neq 0$, означавају исту праву. Зато права не може бити дефинисана коришћењем само једне једначине, већ права мора бити дефинисана као класа једначина које имају пропорционалне коефицијенте. Формализација у систему *Isabelle/HOL* се састоји из неколико корака. Прво, дефинише се домен валидних тројки који су коефицијенти једначине.

typedef `line_coeffs`^{ag} $=$
"$\{((A :: real), (B :: real), (C :: real)).\ A \neq 0 \vee B \neq 0\}$"

Када је овај тип дефинисан, функција `Rep_line_coeffs` ($\lfloor\_\rfloor_{R3}$) конвертује апстрактне вредности овог типа у њихове конкретне репрезентације (тројке реалних бројева), а функција `Abs_line_coeffs` ($\lceil\_\rceil^{R3}$) конвертује (валидне) тројке у вредности које припадају овом типу.

Две тројке су еквивалентне ако и само ако су пропорционалне.

definition "$l_1 \approx^{ag} l_2 \longleftrightarrow$
$(\exists\ A_1\, B_1\, C_1\, A_2\, B_2\, C_2.$
$\lfloor l_1 \rfloor_{R3} = (A_1, B_1, C_1))\ \wedge\ \lfloor l_2 \rfloor_{R3} = (A_2, B_2, C_2)\ \wedge$
$(\exists k.\ k \neq 0 \wedge A_2 = k \cdot A_1 \wedge B_2 = k \cdot B_1 \wedge C_2 = k \cdot C_1))$"

Потом је доказано да је ово релација еквиваленције. Дефиниција за тип праве користи подршку за количничке типове и количничке дефиниције. Значи права (тип $\texttt{line}^{ag}$) се дефинише коришћењем `quotient_type` команде као класа еквиваленције над релацијом $\approx^{ag}$.

Да би избегли коришћење теорије скупова, геометријске аксиоматизације које експлицитно разматрају праве користе релацију инциденције. Ако се користи претходна дефиниција за праву, онда се проверавање инциденције своди на израчунавање да ли тачка (x, y) задовољава једначину праве $A \cdot x + B \cdot y + C = 0$, за неке коефицијенте A, B, и C који су представници класе.

```
definition "ag_in_h (x, y) l ⟷
  (∃ A B C. ⌊l⌋_R3 = (A, B, C) ∧ (A · x + B · y + C = 0))"
```

Ипак, да би доказали да је релација заснована на представницима класе добро дефинисана, мора бити доказано да ако се изаберу други представници класе, рецимо A', B', и C' (који су пропорционални са A, B, и C), онда важи $A' \cdot x + B' \cdot y + C = 0$. Зато ми у нашој Isabelle/HOL формализацији користимо пакет који подржава рад са количничким типовима (`quotient package`). Онда се $A \in^{ag}_{H} l$ дефинише коришћењем **`quotient_definition`** која се заснива на релацији `ag_in_h`. Лема добре дефинисаности је

```
lemma
  shows "l ≈ l' ⟹ ag_in_h P l = ag_in_h P l'"
```

Афина дефиниција. У афиној геометрији, права се дефинише помоћу фиксне тачке и вектора. Као и тачка, вектор такође може бити записан као пар реалних бројева на следећи начин: **`type_synonym`** $\texttt{vec}^{ag}$ = "$real \times real$". Вектори дефинисани на овај начин чине векторски простор (са природно дефинисаним векторским збиром и скаларним производом). Тачке и вектори се могу сабирати као $(x, y) + (v_x, v_y) = (x + v_x, y + v_y)$. Зато, права се записује као тачка и вектор који је различит од нуле:

```
typedef line_point_vec^ag ="(p ::point^ag, v ::vec^ag). v ≠ (0, 0)}"
```

Ипак, различите тачке и вектори могу заправо одређивати једну те исту праву, па конструкција са количничким типом опет мора бити коришћена.

definition "$l_1 \approx^{ag} l_2 \longleftrightarrow (\exists\, p_1\, v_1\, p_2\, v_2.$
$\lfloor l_1 \rfloor_{R3} = (p_1, v_1) \wedge \lfloor l_2 \rfloor_{R3} = (p_2, v_2) \wedge$
$(\exists km.\ v_1 = k \cdot v_2 \wedge p_2 = p_1 + m \cdot v_1))$"

Доказује се да је ово заиста релација еквиваленције. Тада је могуће дефинисати тип којим се представљају праве (`line`^{ag}) као класа еквиваленције над релацијом $\approx^{ag}$ коришћењем команде `quotient_type`.

Ову дефиницију је било могуће задати и коришћењем детерминанти (чиме би се избегло одређивање коефицијената k и m). Услови би били $|v_1, v_2| = 0$ и $|v_1, p_2 - p_1| = 0$.

Након што се докаже добра дефинисаност, инциденција се дефинише на начин који можете видети у наставку (поново се уопштава подизањем на виши ниво) коришћењем количничког пакета.

definiton "ag_in_h $p\, l \longleftrightarrow (\exists\, p_0\, v_0.\ \lfloor l \rfloor_{R3} = (p_0, v_0) \wedge (\exists k.\ p = p_0 + k \cdot v_0))$"

Још једна могућа дефиниција праве је класа еквиваленције парова различитих тачака. Ми нисмо формализовали овај приступ јер је тривијално изоморфан са афином дефиницијом (разлика тачака је вектор који се појављује у афиној дефиницији).

Изометрије

У синтетичкој геометрији изометрије се уводе коришћењем дефиниције. Рефлексије могу прве да се дефинишу, а онда се друге изометрије могу дефинисати као композиција рефлексија. Ипак, у нашој формализацији Декартове равни, изометрије се користе само као помоћно средство да упросте наше доказе (што ће бити додатно појашњено у одељку 4.2). Зато ми нисмо били заинтересовани да дефинишемо изометрије као примитивне појмове (као што су тачке и подударност) него смо представили аналитичке дефиниције и доказали својства која су потребна за касније доказе.

Транслација је дефинисана преко датог вектора (који није експлицитно дефинисан, већ је представљен као пар реалних бројева). Формална дефиниција у *Isabelle/HOL* систему је једноставна.

definiton "transp^{ag} (v_1, v_2) $(x_1, x_2) = (v_1 + x_1, v_2 + x_2)$"

Ротација је параметризована за реални параметар α (који представља угао ротације), а ми само посматрамо ротације око координатног почетка (остале ротације могу се добити као композиција транслације и ротације око координатног почетка). Користимо основна правила тригонометрије да би добили следећу формалну дефиницију у систему *Isabelle/HOL*.

definition "$\texttt{rotp}^{ag}\ \alpha\ (x, y) = ((\cos\alpha)\cdot x - (\sin\alpha)\cdot y, (\sin\alpha)\cdot x + (\cos\alpha)\cdot y)$"

Такође, централна симетрија се лако дефинише коришћењем координата тачке:

definiton "$\texttt{symp}^{ag}\ (x, y)\ = (-x, -y)$"

Важна особина свих изометрија је својство инваријантности, тј. оне чувају основне релације (као што су *између* и *подударно*).

lemma "$\mathcal{B}_T^{ag}\ A\ B\ C \longleftrightarrow \mathcal{B}_T^{ag}\ (\texttt{transp}^{ag}\ v\ A)\ (\texttt{transp}^{ag}\ v\ B)\ (\texttt{transp}^{ag}\ v\ C)$"
lemma "$AB \cong^{ag} CD \longleftrightarrow$
$(\texttt{transp}^{ag}\ v\ A)(\texttt{transp}^{ag}\ v\ B) \cong^{ag} (\texttt{transp}^{ag}\ v\ C)(\texttt{transp}^{ag}\ v\ D)$"
lemma "$\mathcal{B}_T^{ag}\ A\ B\ C \longleftrightarrow \mathcal{B}_T^{ag}\ (\texttt{rotp}^{ag}\ \alpha\ A)\ (\texttt{rotp}^{ag}\ \alpha\ B)\ (\texttt{rotp}^{ag}\ \alpha\ C)$"
lemma "$AB \cong^{ag} CD \longleftrightarrow$
$(\texttt{rotp}^{ag}\ \alpha\ A)(\texttt{rotp}^{ag}\ \alpha\ B) \cong^{ag} (\texttt{rotp}^{ag}\ \alpha\ C)(\texttt{rotp}^{ag}\ \alpha\ D)$"
lemma "$\mathcal{B}_T^{ag}\ A\ B\ C \longleftrightarrow \mathcal{B}_T^{ag}\ (\texttt{symp}^{ag}\ A\ )\ (\texttt{symp}^{ag}\ B\ )\ (\texttt{symp}^{ag}\ C\ )$"
lemma "$AB \cong^{ag} CD \longleftrightarrow (\texttt{symp}^{ag}\ A\ )(\texttt{symp}^{ag}\ B\ ) \cong^{ag} (\texttt{symp}^{ag}\ C\ )(\texttt{symp}^{ag}\ D\ )$"

Изометрије се пре свега користе да трансформишу тачку у њену канонску позицију (обично транслацијом тачке на y-осу). Следеће леме показују да је то могуће учинити.

lemma "$\exists v.\ \texttt{transp}^{ag}\ v\ P = (0, 0)$"
lemma "$\exists \alpha.\ \texttt{rotp}^{ag}\ \alpha\ P = (0, p)$"
lemma "$\mathcal{B}_T^{ag}\ (0, 0)\ P_1\ P_2 \longrightarrow$
$\exists \alpha\ p_1\ p_2.\ \texttt{rotp}^{ag}\ \alpha\ P_1 = (0, p_1)\ \wedge\ \texttt{rotp}^{ag}\ \alpha\ P_2 = (0, p_2)$"

Изометријске трансформације праве се дефинишу коришћењем изометријских трансформација над тачкама (права се трансформише тако што се трансформишу две њене произвољне тачке).

Коришћење изометријских трансформација

Једна од најважнијих техника која је коришћена за упрошћавање формализације ослањала се на коришћење изометријских трансформација. Ми ћемо покушати да представимо мотивациони разлог за примену изометрија на следећем, једноставном примеру.

Лема која се често користи у доказима је да ако $\mathcal{B}_T^{ag}\ A\ X\ B$ и $\mathcal{B}_T^{ag}\ A\ B\ Y$ онда важи $\mathcal{B}_T^{ag}\ X\ B\ Y$. Представићемо како се ова лема може доказати без коришћења изометријских трансформација и како се доказује када се користе изометријске трансформације. Чак и на овом једноставном примеру, ако применимо директан доказ, без коришћења изометријских трансформација, алгебарски рачун постаје превише комплексан.

Нека важи $A = (x_A, y_A)$, $B = (x_B, y_B)$, и $X = (x_X, y_X)$. Како $\mathcal{B}_T^{ag}\ A\ X\ B$ важи, постоји реалан број k_1, $0 \leq k_1 \leq 1$, такав да $(x_X - x_A) = k_1 \cdot (x_B - x_A)$, и $(y_X - y_A) = k_1 \cdot (y_B - y_A)$. Слично, како $\mathcal{B}_T^{ag}\ A\ B\ Y$ важи, постоји реалан број k_2, $0 \leq k_2 \leq 1$, такав да $(x_B - x_A) = k_2 \cdot (x_Y - x_A)$, и $(y_B - y_A) = k_2 \cdot (y_Y - y_A)$. Онда, може се дефинисати реалан број k са $(k_2 - k_2 \cdot k_1)/(1 - k_2 \cdot k_1)$. Ако $X \neq B$, онда коришћењем комплексних алгебарских трансформација, може се доказати да $0 \leq k \leq 1$, и да $(x_B - x_X) = k \cdot (x_Y - x_X)$, и $(y_B - y_X) = k \cdot (y_Y - y_X)$, и зато $\mathcal{B}_T^{ag}\ X\ B\ Y$ важи. Дегенерисани случај $X = B$ тривијално важи.

Ипак, ако применимо изометријске трансформације, онда можемо претпоставити да $A = (0, 0)$, $B = (0, y_B)$, и $X = (0, y_X)$, и да $0 \leq y_X \leq y_B$. Случај када је $y_B = 0$ тривијално важи. У супротном, $x_Y = 0$ и $0 \leq y_B \leq y_Y$. Зато, $y_X \leq y_B \leq y_Y$, и тврђење важи. Приметимо да у овом случају нису биле потребне велике алгебарске трансформације и доказ се ослања на једноставне особине транзитивности релације $\leq$.

Поредећи претходна два доказа, можемо да видимо како примена изометријских трансформација значајно упрошћава потребна израчунавања и скраћује доказе.

Како је ова техника доста коришћена у нашој формализацији, важно је продискутовати који је најбољи начин да се формулишу одговарајуће леме које оправдавају употребу ове технике и покушати што више аутоматизовати коришћење ове технике. Ми смо применили приступ који је предложио Харисон [78].

Својство P је инваријантно под трансформацијом t акко својство P важи за било које тачке које се добију трансформацијом t од тачака за које је својство

P важило.

definiton "inv P $t \longleftrightarrow (\forall\ A\ B\ C.\ P\ A\ B\ C \longleftrightarrow P\ (tA)\ (tB)\ (tC))$"

Тада, следећа лема се може користити да сведемо тврђење које важи за било које тачке које су колинеарне на тврђење за које разматрамо само тачке на y-оси (можемо изабрати и x-осу уколико нам тако више одговара).

lemma
assumes "$\forall\ y_B\ y_C.\ 0 \leq y_B\ \wedge\ y_B \leq y_C \longrightarrow P\ (0,0)\ (0,y_B)\ (0,y_C)$"
"$\forall v.$ inv P (transp^{ag} v)" "$\forall \alpha.$ inv P (rotp^{ag} α)"
"inv P (symp^{ag})"
shows "$\forall A\,B\,C.\ \mathcal{B}_T^{ag}\ A\ B\ C \longrightarrow\ P\ A\ B\ C$"

Доказ да је неко тврђење инваријантно у односу на изометријску трансформацију највише се ослања на леме у којима се доказује да су релација *између* и релација *подударности* инваријантне у односу на изометријске трансформације.

4.3 Модел аксиоматског система Тарског

Прво ћемо навести аксиоме Тарског. Постоје две основне релације:

- Релација *између*: $Bxyz$ означава да је y *између* x и z.
- Релација *подударно*: $xy \equiv vw$ означава да је xy *подударно* vw, односно да је дужина дужи xy једнака дужини дужи vw.

Аксиоме подударности

A1 Рефлексивност подударности: $xy \equiv yx$.

A2 Идентитет подударности: $xy \equiv zz \rightarrow x = y$.

A3 Транзитивност подударности: $(xy \equiv zu \wedge xy \equiv vw) \rightarrow zu \equiv vw$.

Аксиоме распореда

A4 Идентитет релације *између*: $Bxyx \rightarrow x = y$.

A5 Пашова аксиома: $(Bxuz \wedge Byvz) \rightarrow \exists a.(Buay \wedge Bvax)$.

A6 Аксиома непрекидности:
$\exists a \forall x \forall y.[(\phi(x) \wedge \psi(y)) \rightarrow Baxy] \rightarrow \exists b \forall x \forall y.[(\phi(x) \wedge \psi(y)) \rightarrow Bxby]$.

A7 Аксиома доње димензије: $\exists a \exists b \exists c.[\neg Babc \wedge \neg Bbca \wedge \neg Bcab]$.

Аксиоме подударности и распореда

A8 Аксиома горње димензије:
$(xu \equiv xv \wedge yu \equiv yv \wedge zu \equiv zv \wedge u \neq v) \rightarrow (Bxyz \vee Byzx \vee Bzxy)$.

A9 Еуклидова аксиома има три варијанте:
I: $((Bxyw \wedge xy \equiv yw) \wedge (Bxuv \wedge xu \equiv uv) \wedge (Byuz \wedge yu \equiv zu)) \rightarrow yz \equiv vw$.
II: $Bxyz \vee Byzx \vee Bzxy \vee \exists a.(xa \equiv ya \wedge xa \equiv za)$.
III: $(Bxuv \wedge Byuz \wedge x \neq u) \rightarrow \exists a \exists b.(Bxya \wedge Bxzb \wedge Bavb)$.

A10 Аксиома пет дужи:
$(x \neq y \wedge Bxyz \wedge Bx'y'z' \wedge xy \equiv x'y' \wedge yz \equiv y'z' \wedge xu \equiv x'u' \wedge yu \equiv y'u') \rightarrow zu \equiv z'u'$.

A11 Конструкција дужи: $\exists z.Bxyz \wedge yz \equiv ab$.

Наш циљ у овом поглављу је да докажемо да наше дефиниције Декартове координатне равни задовољавају све аксиоме геометрије Тарског [147]. Основни појмови у геометрији Тарског су само три појма - тачке, (инклузивна) релација *између* (означена са $\mathcal{B}_t(A, B, C)$) и релација *подударности* (коју означавамо са $AB \cong_t CD$). У геометрији Тарског праве нису експлицитно дефинисане и колинеарност се дефинише коришћењем релације *између*

definition "$\mathcal{C}_t(A, B, C) \longleftrightarrow \mathcal{B}_t(A, B, C) \vee \mathcal{B}_t(B, C, A) \vee \mathcal{B}_t(C, A, B)$"

Аксиоме подударности.

Прве три аксиоме Тарског представљају основна својства подударности.

lemma „$AB \cong_t BA$"
lemma „$AB \cong_t CC \longrightarrow \ A = B$"
lemma „$AB \cong_t CD \wedge \ AB \cong_t EF \ \longrightarrow CD \cong_t EF$"

Желимо да докажемо да наша релација $\cong^{ag}$ задовољава својства релације $\cong_t$ која је апстрактно задана са претходним аксиомама (тј. да дате аксиоме важе у нашем моделу Декартове координатне равни). У нашој формализацији, аксиоме геометрије Тарског су формулисане коришћењем локала (`locale`, и доказано је да координатна раван представља интерпретацију тог дефинисаног локала. Како је ово техничка страна формализације у *Isabelle/HOL* систему, ми је нећемо овде дискутовати у више детаља (погледати одељак 2.4). На пример, за прву аксиому, доказ се своди на доказивање тврђења $AB \cong^{ag} BA$. Докази су праволинијски и готово аутоматски (поједностављивањем након развијања дефиниција).

Аксиоме распореда.

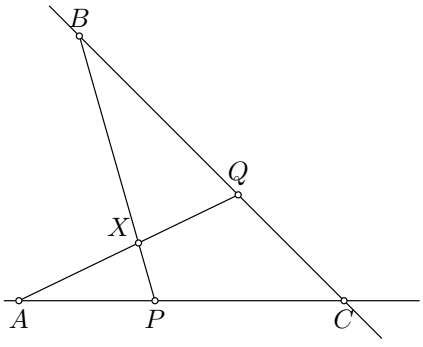
Идентитет у релацији *између*. Прва аксиома (инклузивне) релације *између* даје једно њено једноставно својство и, за наш модел, доказује се готово аутоматски.

`lemma` "$\mathcal{B}_t(A, B, A) \longrightarrow A = B$"

Пашова аксиома. Следећа аксиома је Пашова аксиома:

`lemma` "$\mathcal{B}_t(A, P, C) \wedge \mathcal{B}_t(B, Q, C) \longrightarrow (\exists X.\ (\mathcal{B}_t(P, X, B) \wedge \mathcal{B}_t(Q, X, A)))$"

Под претпоставком да су све тачке које се помињу у аксиоми различите и да нису све тачке колинеарне, слика која одговара аксиоми је:



Пре него што дамо доказ да у нашем моделу Декартове координатне равни важи ова аксиома, желимо да продискутујемо нека питања која се односе на геометрију Тарског и која су се показала важним за свеукупну организацију

нашег доказа. Последња верзија аксиоматског система Тарског је направљена да буде минимална (садржи само 11 аксиома), и централне аксиоме које описују релацију *између* су идентитет релације *између* и Пашова аксиома. У формализацији геометрије Тарског ([128]) сва остала елементарна својства ове релације се изводе из ове две аксиоме. На пример, да би се извела симетричност релације *између* (и.е., $\mathcal{B}_t(A,B,C) \longrightarrow \mathcal{B}_t(C,B,A)$), Пашова аксиома се примењује на тројке ABC и BCC и тада се добија тачка X тако да важи $\mathcal{B}_t(C,X,A)$ и $\mathcal{B}_t(B,X,B)$, и онда према првој аксиоми, $X = B$ и $\mathcal{B}_t(C,B,A)$. Ипак, према нашем искуству, у намери да докажемо да је наша Декартова координатна раван модел аксиома Тарског (поготово за Пашову аксиому), потребно је да већ имамо доказане неке њене последице (као што су симетричност и транзитивност). Додајмо да су раније варијанте аксиоматског система Тарског имале више аксиома, а ова својства су управо била нека од тих додатних аксиома. Такође, својство симетрије је једноставније својство него Пашова аксиома (на пример, оно укључује само тачке које леже на правој, док у аксиоми Паша имамо тачке које леже у равни и не морају бити колинеарне). Додатно, претходни доказ користи веома суптилна својства која зависе од тога како је Пашова аксиома формулисана. На пример, ако се у њеном закључку користи $\mathcal{B}_t(B,X,P)$ и $\mathcal{B}_t(A,X,Q)$ уместо $\mathcal{B}_t(P,X,B)$ и $\mathcal{B}_t(Q,X,A)$, онда доказ не може да се изведе. Зато смо закључили да би добар приступ био да директно докажемо да нека елементарна својства (као што су симетрија и транзитивност) релације *између* важе у моделу, а онда да користимо ове чињенице у доказу много комплексније Пашове аксиоме.

lemma "$\mathcal{B}^{ag}_T\ A\ A\ B$"
lemma "$\mathcal{B}^{ag}_T\ A\ B\ C \longrightarrow \mathcal{B}^{ag}_T\ C\ B\ A$"
lemma "$\mathcal{B}^{ag}_T\ A\ X\ B\ \wedge\ \mathcal{B}^{ag}_T\ A\ B\ Y \longrightarrow \mathcal{B}^{ag}_T\ X\ B\ Y$"
lemma "$\mathcal{B}^{ag}_T\ A\ X\ B\ \wedge\ \mathcal{B}^{ag}_T\ A\ B\ Y \longrightarrow \mathcal{B}^{ag}_T\ A\ X\ Y$"

Пре него што наставимо са доказом да наша Декартова координатна раван у потпуности задовољава Пашову аксиому, потребно је анализирати неколико дегенерисаних случајева. Прва група дегенерисаних случајева настаје када су неке од тачака у конструкцији једнаке. На пример, $\mathcal{B}_t(A,P,C)$ дозвољава да $A = P = C$, или $A = P \neq C$, или $A \neq P = C$ или $A \neq P \neq C$. Директан приступ би био да се сваки од ових случајева посебно анализира. Међутим, бољи приступ је да се пажљиво анализира претпоставка и да се одреди који од

случајева су суштински различити. Испоставља се да су само два различита случаја битна. Ако је $P = C$, онда је Q тражена тачка. Ако је $Q = C$, онда је P тражена тачка. Следећа група дегенерисаних случајева настаје када су све тачке колинеарне. У овом случају важи, или $\mathcal{B}_t(A, B, C)$ или $\mathcal{B}_t(B, A, C)$ или $\mathcal{B}_t(B, C, A)$. У првом случају B је тражена тачка, у другом случају A је тражена тачка, а у трећем случају P је тражена тачка.

Приметимо да се сви дегенерисани случајеви Пашове аксиоме директно доказују коришћењем елементарних својстава и да у овим случајевима није било потребно користити координатна израчунавања. Ово сугерише да су дегенерисани случајеви Пашове аксиоме еквивалентни конјукцији датих својстава. Додатно, ово сугерише да ако се промени аксиоматизација Тарског тако да укључује ова елементарна својства, онда се Пашова аксиома може ослабити тако да садржи само централни случај неколинеарних, различитих тачака.

Коначно, остаје да се докаже централни случај. У том случају, коришћене су алгебарске трансформације да се израчунају координате тачке X и да се докаже претпоставка. Да би се упростио доказ, коришћене су изометрије, као што је описано у одељку 4.2. Почетна конфигурација је трансформисана тако да A постаје координатни почетак, односно $(0, 0)$, да $P = (0, y_P)$ и $C = (0, y_C)$ леже на позитивном делу y-осе. Нека је $B = (x_B, y_B)$, $Q = (x_Q, y_Q)$ и $X = (x_X, y_X)$. Како $\mathcal{B}_t(A, P, C)$ важи, постоји реалан број k_3, $0 \leq k_3 \leq 1$, такав да $y_P = k_3 \cdot y_C$. Слично, како $\mathcal{B}_t(B, Q, C)$ важи, постоји реалан број k_4, $0 \leq k_4 \leq 1$, такав да $(x_B - x_A) = k_2 \cdot (x_Y - x_A)$, и $x_Q - x_B = -k_4 \cdot x_B$ и $y_Q - y_B = k_4 \cdot (y_C - y_B)$. Онда, можемо дефинисати реалан број $k_1 = \frac{k_3 \cdot (1-k_4)}{k_4 + k_3 - k_3 \cdot k_4}$. Како за A, P и C важи $A \neq P \neq C$ и тачке нису колинеарне (јер посматрамо само централни, недегенерисани случај), онда, коришћењем директних алгебарских израчунавања, може бити доказано да $0 \leq k_1 \leq 1$, и да $x_X = k_1 \cdot x_B$, и $y_X - y_P = k_1 \cdot (y_B - y_P)$, и зато $\mathcal{B}_t(P, X, B)$ важи. Слично, можемо дефинисати реалан број $k_2 = \frac{k_4 \cdot (1-k_3)}{k_4 + k_3 - k_3 \cdot k_4}$ и доказати да $0 \leq k_2 \leq 1$ и да важи следеће: $x_X - x_Q = -k_2 \cdot x_Q$ и $y_X - y_Q = -k_2 \cdot y_Q$ и према томе $\mathcal{B}_t(Q, X, A)$ важи. Из ова два закључка ми смо одредили тачку X.

Аксиома ниже димензије. Следећа лема каже да постоје 3 неколинеарне тачке, што представља аксиому ниже димензије у аксиоматици Тарског. Зато сваки модел ових аксиома мора имати димензију већу од 1.

lemma "$\exists\ A\ B\ C.\ \neg\ \mathcal{C}_t(A, B, C)$"

У нашој Декартовој равни тривијално важи (нпр. $(0, 0)$, $(0, 1)$, и $(1, 0)$ су неколинеарне).

Аксиома (схема) непрекидности. Аксиома непрекидности Тарског је у ствари конструкција Дедекиндовог пресека. Интуитивно, ако су све тачке скупа тачака са једне стране у односу на тачке које припадају другом скупу тачака, онда постоји тачка која је између та два скупа. Оригинална аксиоматизација Тарског је дефинисана у оквиру логике првог реда и скупови нису експлицитно познати у оквиру формализације Тарског. Зато, уместо да користи скупове тачака, Тарски користи предикате логике првог реда, ϕ и ψ.

$$(\exists a.\ \forall x.\ \forall y.\ \phi\ x \wedge \psi\ y \longrightarrow \mathcal{B}_t(a, x, y)) \longrightarrow (\exists b.\ \forall x.\ \forall y.\ \phi\ x \wedge \psi\ y \longrightarrow \mathcal{B}_t(x, b, y))$$

Ипак, формулација ове леме у оквиру логике вишег реда система *Isabelle/HOL* не ограничава предикате ϕ и ψ да буду предикати логике првог реда. Зато, строго гледано, наша формализација аксиоматског система Тарског у оквиру система *Isabelle/HOL* даје другачију геометрију у односу на оригиналну геометрију Тарског.

lemma
assumes "$\exists a.\ \forall x.\ \forall y.\ \phi\ x \wedge \psi\ y \longrightarrow \mathcal{B}_T^{ag}\ a\ x\ y$"
shows "$\exists b.\ \forall x.\ \forall y.\ \phi\ x \wedge \psi\ y \longrightarrow \mathcal{B}_T^{ag}\ x\ b\ y$"

Међутим, испоставља се да је могуће доказати да Декартова координатна раван такође задовољава строжију варијанту аксиоме (без ограничавања да предикати ϕ и ψ су предикати логике првог реда). Ако је један скуп празан, тврђење тривијално важи. Ако скупови имају заједничку тачку, онда је та тачка уједно и тражена тачка. У другим случајевим, примењујемо изометријске трансформације тако да све тачке из оба скупа леже на позитивном делу y-осе. Онда, доказ тврђења се своди на доказивање следећег:

lemma
assumes
"$P = \{x.\ x \geq 0 \wedge \phi(0, x)\}$" "$Q = \{y.\ y \geq 0 \wedge \psi(0, y)\}$"
"$\neg(\exists b.\ b \in P \wedge b \in Q)$" "$\exists x_0.\ x_0 \in P$" "$\exists y_0.\ y_0 \in Q$"

"$\forall x \in P.\ \forall y \in Q.\ \mathcal{B}_T^{ag}\ (0,0)\ (0,x)\ (0,y)$"
shows
"$\exists b.\ \forall x \in P.\ \forall y \in Q.\ \mathcal{B}_T^{ag}\ (0,x)\ (0,b)\ (0,y)$"

Доказивање овог тврђења захтева коришћење нетривијалних особина реалних бројева, пре свега, њихову потпуност. Потпуност реалних бројева у систему *Isabelle/HOL* је формализована следећом теоремом (супремум, особина најмање горње границе):

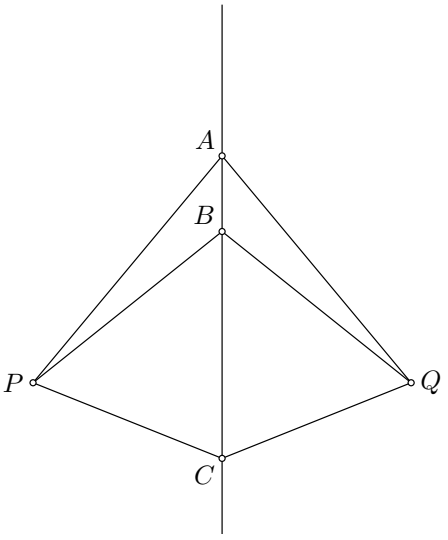
lemma "$(\exists x.\ x \in P) \wedge (\exists y.\ \forall x \in P.\ x < y) \longrightarrow$
$\exists S.\ (\forall y.\ (\exists x \in P.\ y < x) \leftrightarrow y < S)$"

Скуп P задовољава својство супремума. Заиста, како, по претпоставци, P и Q немају заједнички елемент, а из претпоставке следи да $\forall x \in P. \forall y \in Q.\ x < y$, тако да је било који елемент из Q горња граница за P. По претпоставци, P и Q су непразни, тако да постоји елемент b такав да $\forall x \in P.\ x \leq b$ и $\forall y \in Q.\ b \leq y$, а то заправо значи да теорема важи.

Аксиоме подударности и распореда.

Аксиома горње димензије. Три тачке које су на истом одстојању од две различите тачке леже на истој правој. Зато, сваки модел ових аксиома мора имати димензију мању од 3.

lemma "$AP \cong_t AQ \wedge BP \cong_t BQ \wedge CP \cong_t CQ \wedge P \neq Q \longrightarrow \mathcal{C}_t(A,B,C)$"



Ово тврђење је било лако доказати анализом различитих случајева и коришћењем алгебарских трансформација. Није било потребно користити изометријске трансформације.

Аксиома конструкције дужи.

`lemma` "$\exists E.\ \mathcal{B}_t(A, B, E)\ \wedge\ BE \cong_t CD$"

Доказ да наш модел Декартове координатне равни задовољава ову аксиому је једноставан и почиње трансформацијом тачака тако да тачка A постаје координатни почетак, а тачка B лежи на позитивном делу y-осе. Онда $A = (0, 0)$ и $B = (0, b)$, $b \geq 0$. Нека $d = \sqrt{d_{ag}^2\ C\ D}$. Онда $E = (0, b + d)$.

Аксиома пет дужи.

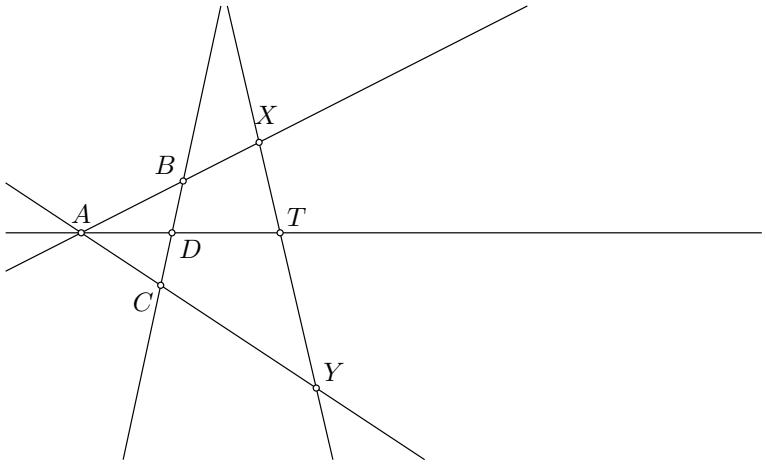
`lemma` "$AB \cong_t A'B' \wedge BC \cong_t B'C' \wedge AD \cong_t A'D' \wedge BD \cong_t B'D' \wedge$
$\mathcal{B}_t(A, B, C) \wedge \mathcal{B}_t(A', B', C') \wedge A \neq B \longrightarrow CD \cong_t C'D'$"

Доказ да наш модел задовољава ову аксиому је прилично директан, али захтева компликована израчунавања. Да бисмо упростили доказ, тачке A, B и C су трансформисане тако да леже на позитивном делу y-осе. Како су у израчунавањима потребни квадратни корени, није било могуће користити аутоматизацију као у претходним доказима и многи ситни кораци су морали бити исписани ручно.

Еуклидова аксиома.

`lemma` "$\mathcal{B}_t(A, D, T) \wedge \mathcal{B}_t(B, D, C) \wedge A \neq D\ \longrightarrow$
$(\exists XY.\ (\mathcal{B}_t(A, B, X)\ \wedge\ \mathcal{B}_t(A, C, Y)\ \wedge\ \mathcal{B}_t(X, T, Y)))$"

Одговарајућа слика када су све тачке различите:



У доказу овог тврђења коришћене су изометријске трансформације. Тачке A, D и T су пресликане редом у тачке $(0,0)$, $(d,0)$ и $(t,0)$, односно у тачке на y-оси. Потом су анализирани дегенерисани случајеви, односно случајеви када су неке од тачака једнаке или када су све тачке колинеарне. У дегенерисаним случајевима, одређивање тачака X и Y није представљало потешкоћу јер углавном су оне неке од датих тачака, односно неке од тачака A, B, C, D или T. Рецимо, уколико су тачке колинеарне и ако важи $\mathcal{B}_t(A,C,T)$, онда је тачка X заправо тачка B, а тачка Y је тачка T.

Доказивање општег случаја захтева доста алгебарских израчунавања. Пре свега, потребно је одредити координате тачака X и Y, а потом на основу тих координата одредити три коефицијента који представљају однос међу тачкама, односно, први коефицијент представља однос међу тачкама A, B и X, други међу тачкама A, C и Y, а трећи међу тачкама X, T и Y. Да бисмо доказали да ове тачке заиста задовољавају релацију $\mathcal{B}_T^{ag}$ _ _ _, потребно је доказати да се сваки од три одређена коефицијента се налази у интервалу $[0,1]$, односно $0 \leq k_i \leq 1$, при чему $i = 1,2,3$. Иако је доказ ове чињенице директан, потребно је доста израчунавања, а због знака $\leq$ није могуће користити аутоматизацију већ је морало да се доста корака доказује ручно.

4.4 Геометрија Хилберта

Прво ћемо навести све аксиоме Хилберта.

Аксиоме инциденције

I–1 За две тачке A, B постоји увек права a која припада свакој од двеју тачака A, B.

I–2 За две тачке A, B не постоји више од једне праве која би припадала свакој од двеју тачака A, B.

I–3 На правој постоје увек најмање две тачке. Постоје најмање три тачке које не леже на једној правој.

I–4 Ма за које три тачке A, B, C, које не леже на истој правој, постоји увек раван α која припада свакој од ове три тачке A, B, C. За сваку раван увек постоји тачка која јој припада.

I–5 За ма које три тачке A, B, C, које не леже на истој правој не постоји више од једне равни која припада свакој од ових трију тачака A, B, C.

I–6 Ако две тачке A, B праве a леже у равни α, онда свака тачка праве a лежи у равни α.

I–7 Ако две равни α, β имају заједничку тачку A, онда оне имају најмање још једну заједничку тачку B.

I–8 Постоје најмање четири тачке које не леже у једној равни.

Аксиоме распореда

II–1 Ако тачка B лежи између тачака A и C, онда су A, B, C три различите тачке праве и B лежи такође између C и A.

II–2 За две тачке A и C увек постоји најмање једна тачка B на правој AB, тако да C лежи између C и A.

II–3 Од ма којих трију тачака праве не постоји више од једне која лежи између оне друге две.

II–4 Нека су A, B, C три тачке које не леже на једној правој и нека је a права у равни ABC која не пролази ни кроз једну од тих тачака; ако дата права пролази кроз једну од тачака дужи AB, она мора пролазити кроз једну од тачака дужи AC, или тачака дужи BC.

Аксиоме подударности

III–1 Ако су A, B две тачке на правој a и ако је, даље, A' тачка на истој или на другој правој a', онда се може увек наћи таква тачка B' праве a' на датој страни од тачке A', да дуж AB буде подударна или једнака дужи $A'B'$, што означавамо на следећи начин: $AB \equiv A'B'$.

III–2 Ако су дужи $A'B'$ и $A''B''$ подударне једној истој дужи AB, биће и дуж $A'B'$ подударна дужи $A''B''$.

III–3 Нека су AB и BC две дужи на правој a без заједничких тачака и нека су, даље, $A'B'$ и $B'C'$ две дужи на истој правој a или на некој другој правој a' које исто тако немају заједничких тачака; ако је тада $AB \equiv A'B'$ и $BC \equiv B'C'$, биће увек и $AC \equiv A'C'$.

III–4 Прво ћемо дати дефиницију угла, а потом и аксиому.

Дефиниција: Нека је α произвољна раван, а h и k нека су ма које различите полуправе које излазе из тачке O у равни α и припадају разним правама. Систем од две полуправе h, k назваћемо *ӯглом* и означаваћемо га са $\angle(h,k)$ или са $\angle(k,h)$.

Аксиома: Нека је дат угао $\angle(h,k)$ у равни α и права a' у равни α' као одређена страна равни α' према правој a'. Нека h' означава полуправу праве a' која полази из тачке O'; онда у равни α' постоји једна и само једна полуправа k' тако да је угао $\angle(h,k)$ подударан или једнак углу $\angle(h',k')$ и у исто време све унутрашње тачке угла $\angle(h',k')$ налазе се на датој страни од праве a', што ћемо означити на овај начин $\angle(h,k) \equiv \angle(h',k')$. Сваки је угао подударан самом себи.

Аксиома паралелности

IV Еуклидова аксиома: Нека је a произвољна права и A тачка ван a: тада постоји у равни, одређеној правом a и тачком A, највише једна права која пролази кроз A и не пресеца a.

Аксиоме непрекидности

V–1 Архимедова аксиома: Ако су AB и CD ма које две дужи, онда постоји такав број n, да када се дуж CD пренесе n од A једно за другим по полуправој која пролази кроз тачку B прелази се преко тачке B.

V–2 Аксиома линеарне потпуности: Систем тачака неке праве са својим релацијама распореда и подударности не може се тако проширити, да остану очуване релације које постоје између претходних елемената као и основне особине линеарног распореда и подударности које проистичу из аксиома I–III, и аксиоме V–1.

Циљ у овом одељку је да докажемо да наше дефиниције Декартовог координатног система задовољавају аксиоме Хилбертове геометрије. Основни објекти у Хилбертовој планарној геометрији су тачке, праве, релација *између* (означена са $\mathcal{B}_h(A, B, C)$) и релација подударности (означена са $AB \cong_h C$).

У оригиналној Хилбертовој аксиоматизацији [81] неке претпоставке се имплицитно подразумевају у односу на контекст у коме су дате. На пример, ако је речено *"постоје две тачке"*, то увек значи постоје две различите тачке. Без ове претпоставке нека тврђења не важе (нпр. релација *између* не важи ако су тачке једнаке).

Аксиоме инциденције

Прве две аксиоме су формализоване коришћењем само једног тврђења.

`lemma` "$A \neq B \longrightarrow \exists!\ l.\ A \in_h l \wedge B \in_h l$"

Последња аксиома ове групе је формализована коришћењем два одвојена тврђења.

`lemma` "$\exists AB.\ A \neq B \wedge A \in_h l \wedge B \in_h l$"
`lemma` "$\exists ABC.\ \neg\ \mathcal{C}_h(A, B, C)$"

Релација колинеарности $\mathcal{C}_h$ (која је коришћена у претходној дефиницији) се дефинише на следећи начин:

`definition` "$\mathcal{C}_h(A, B, C) \longleftrightarrow \exists l.\ A \in_h l\ \wedge\ B \in_h l\ \wedge\ C \in_h l$."

Наравно, ми желимо да докажемо да наш модел (са нашим дефиницијама тачке, праве и основних релација (подударно, између)) задовољава ове аксиоме. На пример, ово значи да ми треба да докажемо:

lemma „$A \neq B \longrightarrow \exists l.\ A \in_H^{ag} l \wedge B \in_H^{ag} l$."

Докази ових лема су тривијални и углавном су добијени развијањем дефиниција и онда коришћењем аутоматског доказивања (коришћењем методе Гребнерових база).

Аксиоме поретка

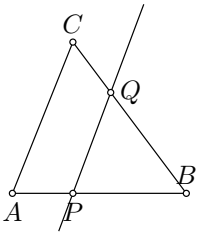
Аксиоме поретка описују својства (ексклузивне) релације *између*.

lemma "$\mathcal{B}_h(A,B,C) \longrightarrow A \neq B \wedge A \neq C \wedge B \neq C \wedge \mathcal{C}_h(A,B,C) \wedge \mathcal{B}_h(C,B,A)$"
lemma "$A \neq C \longrightarrow \exists B.\ \mathcal{B}_h(A,C,B)$"
lemma "$A \in_h l \wedge B \in_h l \wedge C \in_h l \wedge A \neq B \wedge B \neq C \wedge A \neq C \longrightarrow$
$(\mathcal{B}_h(A,B,C) \ \wedge \ \neg\mathcal{B}_h(B,C,A) \ \wedge \ \neg\mathcal{B}_h(C,A,B)) \ \vee$
$(\neg\mathcal{B}_h(A,B,C) \ \wedge \ \mathcal{B}_h(B,C,A) \ \wedge \ \neg\mathcal{B}_h(C,A,B)) \ \vee$
$(\neg\mathcal{B}_h(A,B,C) \ \wedge \ \neg\mathcal{B}_h(B,C,A) \ \wedge \ \mathcal{B}_h(C,A,B))$"

Докази да релације $\cong^{ag}$, $\in_H^{ag}$, и $\mathcal{B}_H^{ag}$ задовољавају ове аксиоме су једноставни и углавном су изведени одмотавањем дефиниција и коришћењем аутоматизације.

Пашова аксиома.

lemma "$A \neq B \wedge B \neq C \wedge C \neq A \wedge \mathcal{B}_h(A,P,B) \wedge$
$P \in_h l \wedge \neg C \in_h l \wedge \neg A \in_h l \wedge \neg B \in_h lh \ \longrightarrow$
$\exists Q.\ (\mathcal{B}_h(A,Q,C) \ \wedge \ Q \in_h l) \ \vee \ (\mathcal{B}_h(B,Q,C) \ \wedge \ Q \in_h l)$"



У оригиналној Пашовој аксиоми постоји још једна претпоставка – тачке A, B и C нису колинеарне, тако да је аксиома формулисана само за централни, недегенерисани случај. Ипак, у нашем моделу тврђење тривијално важи ако оне јесу колинеарне, тако да смо ми доказали да наш модел задовољава и

централни случај и дегенерисани случај када су тачке колинеарне. Приметимо да због својстава Хилбертове релације *између*, претпоставка да су тачке различите не може бити изостављена.

Доказ користи стандардне технике. Прво, користе се изометријске трансформације да транслирају тачке на y-оси, тако да $A = (0, 0)$, $B = (x_B, 0)$ и $P = (x_P, 0)$. Нека је $C = (x_C, y_C)$ и $\lfloor l \rfloor_{R3} = (l_A, l_B, l_C)$. У зависности у којим дужима се тражена тачка налази, имамо два велика различита случаја. Коришћењем својства $\mathcal{B}_h(A, P, B)$ доказује се да важи $l_A \cdot y_B \neq 0$ и онда можемо одредити два коефицијента $k_1 = \frac{-l_C}{l_A \cdot y_B}$ и $k_2 = \frac{l_A \cdot y_B + l_C}{l_A \cdot y_B}$. Даље, доказује се да важи $0 < k_1 < 1$ или $0 < k_2 < 1$. Коришћењем $0 < k_1 < 1$, тачка $Q = (x_Q, y_Q)$ је одређена са $x_Q = k_1 \cdot x_C$ и $y_Q = k_1 \cdot y_C$, па зато $\mathcal{B}_h(A, Q, C)$ важи. У другом случају, када друго својство важи, тачка $Q = (x_q, y_q)$ је одређена са $x_Q = k_2 \cdot (x_C - x_B) + x_B$ и $y_Q = k_2 \cdot y_C$, па зато $\mathcal{B}_t(B, Q, C)$ важи.

Аксиоме подударности

Прва аксиома подударности омогућава конструисање подударних дужи на датој правој. У Хилбертовој књизи „Основи геометрије" [81] аксиома се формулише на следећи начин: *„Ако су A и B две тачке на правој a, A' је тачка на истој или другој правој a' онда је увек могуће одредити тачку B' на датој страни праве a' у односу на тачку A' такву да је дуж AB подударна дужи $A'B'$."* Ипак, у нашој формализацији део *„на датој страни"* је промењен и уместо једне одређене су две тачке (приметимо да је ово имплицитно и речено у оригиналној аксиоми).

lemma "$A \neq B \wedge A \in_h l \wedge B \in_h l \wedge A' \in_h l' \longrightarrow$
$\exists B'\, C'.\ B' \in_h l' \wedge C' \in_h l' \wedge \mathcal{B}_h(C', A', B') \wedge AB \cong_h A'B' \wedge AB \cong_h A'C'$"

Доказ да ова аксиома важи у нашем моделу Декартове координатне равни, почиње са изометријским трансформацијама тако да A' постаје $(0, 0)$ и l' постаје x-оса. Тада је прилично једноставно одредити две тачке на x-оси тако што одредимо координате ових тачака користећи услов да d^2_{ag} (квадратно растојање) између било које од њих и тачке A' је исто као и d^2_{ag} A B.

Следеће две аксиоме су директно доказане одвијањем одговарајућих дефиниција и применом алгебарских трансформација и метода Гребнерових база.

lemma "$AB \cong_h A'B' \wedge AB \cong_h A''B'' \longrightarrow A'B' \cong_h A''B''$"

lemma "$\mathcal{B}_h(A,B,C) \wedge \mathcal{B}_h(A',B',C') \wedge AB \cong_h A'B' \wedge BC \cong_h B'C' \longrightarrow AC \cong_h A'C'$"

Следеће три аксиоме подударности у Хилбертовој аксиоматизацији су о појму угла, а ми у оквиру ове тезе нећемо разматрати формализацију угла.

За разлику од осталих аксиома које су формулисане на основу основних појмова (тачка, права, *припада*, *подударно*, *између*), у овим аксиомама се јавља потреба коришћења уведених појмова (нпр. углова, полуправих, подударности углова, припадности полуправе правој, ...). Све ове помоћне дефиниције постају саставни део аксиоматског система и он постаје непотребно велики.

Појам угла је сам по себи веома широк и заправо се под једним неформалним термином крију различити математички појмови. Угао се може дефинисати преко две полуправе које се секу, али и преко две праве које се секу. када се фиксирају полуправе, остаје питање који од два дела равни се подразумева под углом – да ли угао увек мора бити конвексан или су те полуправе уређен пар, па се подразумева оријентисани угао од прве праве ка другој. Проблем са Хилбертовим текстом је што иако се може закључити која је мотивација датих дефиниција, на пуно места текст је непрецизан и потребно је извршити одређена прецизирања да би се добила формална дефиниција (а то се може урадити на различите начине и самим тим аксиоматске основе могу бити различите). На пример, није јасно шта значи да тачка припада углу – да ли је то скуп тачака или посебан тип за који је потребно увести нову релацију инциденције. Потом, формално су уведене полуправе, али не и критеријуму када две полуправе сачињавају праву, нити када полуправа припада правој, што представља проблем јер се и ти критеријуми користе (на пример, када се уводе сумплементарни углови).

Аксиома паралелности

lemma "$\neg P \in_h l \longrightarrow \exists!\, l'.\ P \in_h l' \wedge \neg(\exists\ P_1.\ P_1 \in_h l \wedge P_1 \in_h l')$"

Доказ ове аксиоме састоји се из два дела. Прво је доказано да таква права постоји а потом да је она јединствена. Доказивање постојања је учињено одређивањем коефицијената тражене праве. Нека је $P = (x_P, y_P)$ и $\lfloor l \rfloor_{R3} = (l_A, l_B, l_C)$. Онда су коефицијенти тражене праве $(l_A, l_B, -l_A \cdot x_P - l_B \cdot y_P)$.

У другом делу доказа, полази се од претпоставке да постоје две праве које задовољавају услов $P \in_h l' \ \wedge \ \neg(\exists\ P_1.\ P_1 \in_h l \ \wedge \ P_1 \in_h l')$. Доказано је да су њихови коефицијенти пропорционални па су самим тим и праве једнаке.

Аксиоме непрекидности

Архимедова аксиома. Нека је A_1 нека тачка на правој између случајно изабраних тачака A и B. Нека су тачке $A_2, A_3, A_4, \ldots$ такве да A_1 лежи између тачке A и A_2, A_2 између A_1 и A_3, A_3 између A_2 и A_4 итд. Додатно, нека су дужи $AA_1, A_1A_2, A_2A_3, A_3A_4, \ldots$ једнаки међусобно. Онда, у овом низу тачака, увек постоји тачка A_n таква да B лежи између A и A_n.

Прилично је тешко репрезентовати низ тачака на начин како је то задато у аксиоми и наше решење је било да користимо листу. Прво, дефинишемо листу такву да су сваке четири узастопне тачке подударне, а за сваке три узастопне тачке важи релација *између*.

```
definition
  "congruentl l ⟶ length l ≥ 3 ∧
          ∀i. 0 ≤ i ∧ i + 2 < length l ⟶
          (l ! i)(l ! (i + 1)) ≅h (l ! (i + 1))(l ! (i + 2)) ∧
          ℬh((l ! i), (l ! (i + 1)), (l ! (i + 2)))"
```

Са оваквом дефиницијом, аксиома је мало трансформисана, али и даље са истим значењем, и она каже да постоји низ тачака са својствима која су горе поменута таква да за барем једну тачку A' из дате листе важи $\mathcal{B}_t(A, B, A')$. У *Isabelle/HOL* систему ово је формализовано на следећи начин:

```
lemma "ℬh(A, A1, B) ⟶
  (∃l. congruentl (A # A1 # l) ∧ (∃i. ℬh(A, B, (l ! i))))"
```

Главна идеја овог доказа је у тврђењима $d_{ag}^2\ A\ A' > d_{ag}^2\ A\ B$ и $d_{ag}^2\ A\ A' = t \cdot d_{ag}^2\ A\ A_1$. Зато, у првом делу доказа одредимо t такво да $t \cdot d_{ag}^2\ A\ A_1 > d_{ag}^2\ A\ B$ важи. Ово се постиже применом Архимедовог правила за реалне бројеве. Даље, доказано је да постоји листа l таква да `congruentl` l важи, да је та листа дужа од t, и таква да су њена прва два елемента A и A_1. Ово је урађено индукцијом по параметру t. База индукције, када је $t = 0$ тривијално важи. У индукционом кораку, листа је проширена са једном тачком таквом да важи

релација подударности за њу и последње три тачке листе и да важи релација *између* за последња два елемента листе и додату тачку. Коришћењем ових услова, координате нове тачке се лако одређују алгебарским израчунавањима. Када је једном конструисана, листа задовољава услове аксиоме, што се лако доказује у последњим корацима доказа. У доказу се користе неке додатне леме које углавном служе да се опишу својства листе која задовољава услов `congruentl` l.

4.5 Завршна разматрања

У овој тези ми смо представили добро изграђену формализацију Декартове геометрије равни у оквиру система *Isabelle/HOL*. Дато је неколико различитих дефиниција Декартове координатне равни и доказано је да су све дефиниције еквивалентне. Дефиниције су преузете из стандардних уџбеника. Међутим, да би их исказали у формалном окружењу асистента за доказивање теорема, било је потребно подићи ниво ригорозности. На пример, када дефинишемо праве преко једначина, неки уџбеници помињу да различите једначине репрезентују исту праву ако су њихови коефицијенти "пропорционални", док неки други уџбеници често ово важно тврђење и не наведу. У текстовима се обично не помињу конструкције као што су релација еквиваленције и класа еквиваленције које су у основи наше формалне дефиниције.

Формално је доказано да Декартова координатна раван задовољава све аксиоме Тарског и већину аксиома Хилберта (укључујући и аксиому непрекидности). Доказ да наша дефиниција Декартове координатне равни задовољава све аксиоме Хилберта је тема за наредни рад јер смо констатовали да формулација аксиоме комплетности и аксиома у којима се помиње изведени појам угла представљају велики изазов за формализацију јер су у Хилбертовом тексту задате веома непрецизно.

Наше искуство је да доказивање да наш модел задовољава једноставне Хилбертове аксиоме лакше него доказивање да модел задовољава аксиоме Тарског. Разлог за ово највише лежи у дефиницији релације *између*. Наиме, Тарски дозвољава да тачке које су у релацији *између* буду једнаке. Ово је разлог за постојање бројних дегенерисаних случајева који морају да се анализирају посебно што додатно усложњава расуђивање и доказе. Међутим, Хилбертове аксиоме су формулисане коришћењем неких изведених појмова

(нпр. углова) што представља проблем за нашу формализацију.

Чињеница да је аналитичка геометрија модел синтетичке геометрије се често подразумева као једна једноставна чињеница. Ипак, наше искуство показује да, иако концептуално једноставан, доказ ове чињенице захтева прилично комплексна израчунавања и веома је захтеван за формализацију. Испоставља се да је најважнија техника коришћена да се упросте докази "без губитка на општости" и коришћење изометријских трансформација. На пример, прво смо покушали да докажемо централни случај Пашове аксиоме без примене изометријских трансформација. Иако би требало да буде могуће извести такав доказ, израчунавања која су се појавила су била толико комплексна да ми нисмо успели да завршимо доказ. После примене изометријских трансформација, израчунавања су и даље била нетривијална, али ипак, ми смо успели да завршимо овај доказ. Треба имати на уму да смо морали да се често користимо ручним израчунавањима јер чак и моћна тактика која се заснива на Гребенеровим базама није успела да аутоматски упрости алгебарске изразе. Из овог експеримента са Пашовом аксиомом, закључили смо колики је значај изометријских трансформација и следећа тврђења нисмо ни покушавали да доказујемо директно.

Наша формализација аналитичке геометрије се заснива на аксиомама реалних бројева и у многим доказима су коришћена својства реалних бројева. Многа својства важе за било које поље бројева (и тактика заснована на Гребенеровим базама је такође била успешна и у том случају). Међутим, да би доказали аксиому непрекидности користили смо својство супремума, које не важи у произвољном пољу. У нашем даљем раду, планирамо да изградимо аналитичку геометрију без коришћења аксиома реалних бројева, тј. да дефинишемо аналитичку геометрију у оквиру аксиоматизације Тарског или Хилберта. Заједно са овим радом, то би омогућило дубљу анализу неких теоријских својстава модела геометрије. На пример, желимо да докажемо категоричност и система аксиома Тарског и система аксиома Хилберта (и да докажемо да су сви модели изоморфни и еквивалентни Декартовој координатној равни).

Глава 5

Формализација хиперболичке геометрије

5.1 Увод

Постоји много радова и књига које описују геометрију комплексне равни, а у овом поглављу ми ћемо представити резултате наше формализације. Постоји више циљева које смо желели да остваримо.

1. Формализовати теорију проширене комплексне равни (комплексна раван која садржи тачку бесконачно) и њених објеката (правих и кругова) и њених трансформација (на пример, инверзија и Мебијусових трансформација).

2. Спојити бројне приступе које можемо срести у препорученој литератури ([130], [148], [117]) у један униформни приступ у коме ће бити коришћен јединствен и прецизан језик за описивање појмова.

3. Анализирати и формално доказати све случајеве који често остану недовољно истражени јер их више различитих аутора сматра тривијалним.

4. У раду ће бити детаљно дискутовани односи између два приступа у формализацији као и њихове предности и мане. Природно се намећу два приступа формализацији: геометријски (рецимо приступ који предлаже Нидам [130]) и алгебарски (приступ који можемо видети у раду Шверд-фегера [148]), као и питање да ли избор приступа утиче на ефикасност формалног доказивања.

5. Анализирати технике које се користе у доказима, као и могућност коришћења аутоматизације.

6. Посматрати да ли је доказе лакше извести у моделу Риманове сфере или у моделу хомогених координата.

У овој тези, ради сажетости, ми ћемо представити само основне резултате наше формализације — најважније дефиниције и тврђења. Ова теза садржи само кратку рекапитулацију оригиналног формалног извођења и многа својства која су формално доказана неће бити презентована у овом раду. Додатно, ни један доказ неће бити приказан или описан, а све је доступно у оквиру званичне *Isabelle/HOL* документације [1]. У тези ће бити описане технике које смо користили, као и леме и помоћна тврђења која су била потребна у доказима, али докази неће бити комплетно приказани јер су превише технички, често веома велики и читаоцу могу бити неинтересантни.

Организација поглавља. У поглављу 5.2 дате су дефиниције основних појмова који се користе – комплексни бројеви, дводимензиони вектори и аритметичке операције са њима, матрице димензије 2×2 и основне аритметичке операције са матрицама, као и дефиниције адјунговане, хермитске и унитарне матрице. У поглављу 5.3 дата је дефиниција хомогених координата којима се представљају комплексни бројеви проширене комплексне равни, потом су дефинисане аритметичке операције над хомогеним координатама и дате су леме о њиховим својствима, а потом су дефинисане размера и дворазмера и доказана су основна својства ових операција. У поглављу 5.4 успостављена је веза између Риманове сфере и проширене комплексне равни коришћењем стереографске пројекције. Ова веза је важна јер се често показало лакшим доказати тврђење на Римановој сфери него у проширеној комплексној равни. Такође, уведена је и метрика коришћењем тетивног растојања. У поглављу 5.5 дефинисане су Мебијусове трансформације и дата је њихова карактеризација коришћењем матрица димензије 2×2. Доказано је да Мебијусове трансформације формирају групу над композицијом. Потом је дефинисано дејство Мебијусових трансформација на тачке проширене комплексне равни. Као значајна група Мебијусових трансформација издвајају се еуклидске сличности које такође формирају групу и доказано је да се свака еуклидска сличност

[1] *Isabelle* документи у којима су теорије и докази доступни се налазе на адреси `http://www.matf.bg.ac.rs/~danijela/Moebius.zip`

може добити композицијом ротације, транслације и хомотетије. Једно од најзначајнијих тврђења је да је дворазмера такође Мебијусова трансформација. Ово својство је коришћено да се докаже да се Мебијусовом трансформацијом било које три тачке могу сликати у било које три тачке. Доказивањем те чињенице, многи докази су олакшани јер се тврђење могло свести на доказивање да оно важи за неке специјално изабране тачке, односно коришћењем механизма „без губитка на општости". У поглављу 5.6 уведен је појам уопштеног круга (кругоправа) којом се у проширеној комплексној равни могу представити и еуклидски кругови, али и еуклидске праве. Успостављена је веза између равни у простору и кругоправих коришћењем стереографске пројекције. Доказано је да се Мебијусовим трансформацијама кругоправа слика у кругоправу. Потом је доказано да се карактеризација кругоправих може поделити на три врсте: имагинарне кругоправе (које немају тачака), тачка кругоправе (имају једну тачку) и реалне кругоправе (имају барем три тачке). Уведена је оријентација кругоправих и тиме је омогућено дефинисање диска, односно унутрашњости кругоправе. Доказана су и тврђења која се односе на дејство Мебијусових трансформација на оријентисане кругоправе. Једно од најзначајнијих тврђења је да Мебијусове трансформације чувају угао између кругоправих. У поглављу 5.7 дата је класификација Мебијусових трансформација. Најзначајнији су аутоморфизми диска који сликају унутрашњост диска у унутрашњост диска. Такође, доказано је да се у односу на фиксне тачке, Мебијусове трансформације могу класификовати у неколико група у којима све трансформације имају заједничка својства. У поглављу 5.8 описан је геометријски приступ у доказивању да Мебијусова трансформација чува угао и дискутована је разлика између алгебарског и геометријског приступа у доказу. У поглављу 5.10 посматра се јединични диск и потребно је доказати да он представља модел геометрије Лобачевског. Потребно је доказати да у њему важе све аксиоме Тарског, осим Еуклидове аксиоме. Дефинисана је релација између и доказана су њена основна својства, на пример, како Мебијусова трансформација утиче на релацију између. Потом је доказано да неке аксиоме Тарског важе на овом диску. На крају су сумирани сви закључци ове главе.

5.2 Основни појмови геометрије комплексне равни

Комплексни бројеви. Иако у систему *Isabelle/HOL* постоји основна подршка за комплексне бројеве (са `complex` је означен тип комплексних бројева у систему *Isabelle/HOL*), то није било довољно за наше потребе, па смо морали да направимо додатни напор и да ову теорију проширимо. Многе леме које смо доказали су углавном веома техничке и нису интересантне за виши ниво формализације коју описујемо и зато их нећемо спомињати у овом тексту (нпр. **lemma** "arg i = $pi/2$" или **lemma** "$|z|^2$ = Re (z * cnj z)"). Једна од најзначајнијих дефиниција је дефиниција функције за канонизацију угла $\lfloor \_ \rfloor$, која узима у обзир 2π периодичност функција sin и cos и пресликава сваки угао у његову каноничну вредност која припада интервалу $(-\pi, \pi]$. Са овом функцијом, на пример, мултипликативна својства функције arg могу се лако изразити и доказати.

lemma "$z_1 * z_2 \neq 0 \Longrightarrow$ arg($z_1 * z_2$) = $\lfloor$arg z_1 + arg $z_2\rfloor$"

Како се комплексни бројеви често третирају и као вектори, увођење скаларног производа два комплексна броја (што је дефинисано као $\langle z_1, z_2\rangle = (z_1 *$ cnj $z_2 + z_2 *$ cnj $z_1)/2$) се показало веома корисним за сажето приказивање неких услова.

Линеарна алгебра. Следећа важна теорија за даљу формализацију је теорија линеарне алгебре $\mathbb{C}^2$. Представљање вектора и матрица различитих димензија у логици вишег реда представља изазов, због недостатка зависних типова [76], али у нашој формализацији треба само да разматрамо просторе коначне димензије $\mathbb{C}^2$ и у неким ситуацијама $\mathbb{R}^3$, тако да је наш задатак био једноставнији. Комплексни вектори се дефинишу са **type_synonym** C2_vec = complex × complex. Слично, комплексне матрице (C2_mat) се дефинишу као четворка комплексних бројева (матрица $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ репрезентована је са (A, B, C, D)). Скаларно множење вектора означавамо са $*_{sv}$, а скаларно множење са матрицом означавамо са $*_{sm}$. Скаларни производ два вектора означен је са $*_{vv}$, производ вектора и матрице је означено са $*_{vm}$, производ матрице и вектора је означено са $*_{mv}$, а производ две матрице са $*_{mm}$. Нула матрица је означена са mat_zero, јединична матрица је означена са eye, нула вектор

је означен са `vec_zero`, детерминанта матрице је означена са `mat_det`, њен траг (сума елемената на главној дијагонали) са `mat_trace`, инверзна матрица са `mat_inv`, транспонована матрица са `mat_transpose`, коњугација сваког елемента вектора са `vec_cnj`, коњугација сваког елемента матрице са `mat_cnj`, итд. Уведени су многи стандардни појмови линеарне алгебре. На пример, сопствене вредности су дефинисане и окарактерисане на следећи начин:

```
definition eigenval :: "C2_mat ⇒ complex ⇒ bool" where
  "eigenval k A ⟷ (∃v. v ≠ vec_zero ∧ A *mv v = k *sv v)"
lemma "eigenval k H ⟷ k^2 − mat_trace H * k + mat_det H = 0"
```

Адјунгована матрица је транспонована коњугована матрица. *Хермитске* матрице су оне које су једнаке својој адјунгованој матрици, док су *унитарне* матрице оне чији инверз је једнак њиховој адјунгованој матрици.

```
definition mat_adj where "mat_adj H = mat_cnj (mat_transpose H)"
definition hermitean where "hermitean H ⟷ mat_adj H = H"
definition unitary where "unitary M ⟷ mat_adj M *mm M = eye"
```

Други основни појмови који су потребни у овом раду ће бити уведени у даљем тексту, а читалац може пронаћи више информација у нашој формализацији [2].

Проширена комплексна раван

Веома важан корак у развоју геометрије комплексне равни је проширена комплексна раван која има један додатни елемент у односу на комплексну раван $\mathbb{C}$ (који се третира као тачка бесконачно). Проширену комплексну раван ћемо означити са $\overline{\mathbb{C}}$. Постоји више различитих приступа [130, 148] за дефинисање $\overline{\mathbb{C}}$. Најпривлачнији начин са становишта израчунавања је приступ који се базира на хомогеним координатама, а најпривлачнији приступ визуелно је заснован на стереографској пројекцији Риманове сфере.

5.3 Хомогене координате

Проширена комплексна раван $\overline{\mathbb{C}}$ се идентификује са комплексном пројективном правом (једнодимензиони пројективни простор над комплексним по-

[2] `http://www.matf.bg.ac.rs/~danijela/Moebius.zip`

љем, понекад означаван са $\mathbb{C}P^1$). Свака тачка $\overline{\mathbb{C}}$ је репрезентована паром комплексних хомогених координата (од којих нису оба једнака нули), а два пара хомогених координата представљају исту тачку у $\overline{\mathbb{C}}$ акко су они пропорционални са неким ненула комплексним фактором. Формализација овог својства у систему *Isabelle/HOL* се ослања на *lifting/transfer* пакет за колинички тип [84] и састоји се из три фазе [3]. Прво се уводи тип за пар комплексних бројева који је различит од нуле (и који се истовремено посматра и као ненула комплексни вектор).

```
typedef C2_vec≠0 = "{v::C2_vec. v ≠ vec_zero}"
```

Одавде добијамо функцију за репрезентацију `Rep_C2_vec`$_{\neq 0}$ (коју ћемо означити са $\lfloor \_ \rfloor_{C2}$) која враћа (ненула) пар комплексних бројева за сваки дати елемент помоћног типа `C2_vec`$_{\neq 0}$ и враћа функцију за апстракцију `Abs_C2_vec`$_{\neq 0}$ (коју ћемо ми означити са $\lceil \_ \rceil^{C2}$) која враћа `C2_vec`$_{\neq 0}$ елемент за сваки дати не-нула пар комплексних бројева. Друго, кажемо да су два елемента типа `C2_vec`$_{\neq 0}$ еквивалентна акко су њихове репрезентације пропорционалне.

```
definition ≈C2 :: "C2_vec≠0 ⇒ C2_vec≠0 ⇒ bool" where
  "z1 ≈C2 z2 ⟷ (∃ (k::complex). k ≠ 0 ∧ ⌊z2⌋C2 = k *sv ⌊z1⌋C2)"
```

Било је могуће увести и нешто другачију дефиницију, односно да се уместо постојања параметра k испитује вредност детерминанте, односно да ли је $|\lfloor z_1 \rfloor_{C2}, \lfloor z_2 \rfloor_{C2}| = 0$.
Прилично је лако доказати да је $\approx_{C2}$ релација еквиваленције. Коначно, тип комплексних бројева проширене комплексне равни дат хомогеним координатама се дефинише као класа еквиваленције релације $\approx_{C2}$ и уводи се преко наредног количничког типа.

```
quotient_type complexhc = C2_vec≠0 / ≈C2
```

Да сумирамо, на најнижем нивоу репрезентације постоји тип комплексних бројева, на следећем нивоу је тип ненула комплексних 2×2 вектора (који се представљају претходним типом), а на највишем нивоу је количнички тип

[3] Једна од фаза може бити прескочена ако би се користио *lifting/transfer* пакет за парцијални количнички тип. Ову могућност ми нисмо користили у нашој формализацији због неких проблема који су постојали у ранијим верзијама количничког пакета. У међувремену, сви проблеми су исправљени, али наша формализације је у том тренутку већ увелико била развијена.

који има класу еквиваленије — рад са овим количничким типом (његовом репрезентацијом и апстракцијом) се ради у позадини, коришћењем пакета *lifting/transfer* [84]. Ова три нивоа апстракције могу на математичаре деловати збуњујуће, али они су неопходни у формалном окружењу где сваки објекат мора имати јединствени тип (на пример, често се узима да је $(1, i)$ истовремено пар комплексних бројева и ненула комплексни вектор, али у нашој формализацији $(1, i)$ је пар комплексних бројева, док је $\lceil(1, i)\rceil^{C2}$ ненула комплексни вектор).

Обични и бесконачни бројеви. Сваки обични комплексни број може бити конвертован у проширени комплексни број.

```
definition of_complex_rep :: "complex ⇒ C2_vec≠0" where
  of_complex_rep z = ⌈(z, 1)⌉^C2
lift_definition of_complex :: "complex ⇒ complex_hc" is
  of_complex_rep
```

Тачка бесконачности се дефинише на следећи начин:

```
definition inf_hc_rep :: C2_vec≠0 where inf_hc_rep = ⌈(1, 0)⌉^C2
lift_definition ∞_hc :: "complex_hc" is inf_hc_rep
```

Лако се доказује да су сви проширени комплексни бројеви или ∞_{hc} (акко је њихова друга координата једнака нули) или се могу добити конвертовањем обичних комплексних бројева (акко њихова друга координата није нула).

```
lemma "z = ∞_hc ∨ (∃ x. z = of_complex x)"
```

Нотација 0_{hc}, 1_{hc} и i_{hc} се користи да означи комплексне бројеве 0, 1, и i у проширеној комплексној равни (у хомогеним координатама).

Аритметичке операције. Аритметичке операције обичних комплексних бројева могу бити проширене тако да се могу применити у проширеној комплексној равни.

На најнижем, репрезентативном нивоу, сабирање (z_1, z_2) и (w_1, w_2) се дефинише као $(z_1 * w_2 + w_1 * z_2, z_2 * w_2)$, тј.

```
definition plus_hc_rep :: "C2_vec≠0 ⇒ C2_vec≠0 ⇒ C2_vec≠0"
  where "plus_hc_rep z w = (let (z1, z2) = ⌊z⌋C2; (w1, w2) = ⌊w⌋C2
                            in ⌈(z1 * w2 + w1 * z2, z2 * w2)⌉C2)"
```

Овим се добија ненула пар хомогених координата осим ако су и z_2 и w_2 нула (у супротном, добија се лоше дефинисани елемент $\lceil(0,0)\rceil^{C2}$)[4]. Дефиниција је подигнута на ниво количничког типа:

```
lift_definition +hc :: "complexhc ⇒ complexhc ⇒ complexhc" is
  plus_hc_rep
```

Ова дефиниција генерише следећи обавезан услов који треба доказати $[\![z \approx_{C2} z'; w \approx_{C2} w']\!] \Longrightarrow z +_{hc} w \approx_{C2} z' +_{hc} w'$, а он се лако доказује анализом случајева да ли су z_2 и w_2 оба једнака нули. Приметимо да због HOL захтева да све функције буду тоталне, ми не можемо дефинисати функцију само за добро дефинисане случајеве, и у доказу морамо да посматрамо и лоше дефинисане случајеве.

Даље, доказује се да ова операција проширује уобичајено сабирање комплексних бројева (операцију + у $\mathbb{C}$).

```
lemma "of_complex z +hc of_complex w = of_complex (z + w)"
```

Сума обичних комплексних бројева и ∞_{hc} је ∞_{hc} (ипак, $\infty_{hc} +_{hc} \infty_{hc}$ је лоше дефинисана).

```
lemma "of_complex z +hc ∞hc = ∞hc"
lemma "∞hc +hc of_complex z = ∞hc"
```

Операција $+_{hc}$ је асоцијативна и комутативна, али ∞_{hc} нема инверзни елемент, што прекида лепа алгебарска својства операције + на $\mathbb{C}$.

Друге аритметичке операције су такође проширене. На најнижем, репрезентативном нивоу, унарни минус (z_1, z_2) је $(-z_1, z_2)$, производ (z_1, z_2) и (w_1, w_2) је $(z_1 * z_2, w_1 * w_2)$, а реципрочна вредност (z_1, z_2) је (z_2, z_1) – ове операције су онда подигнуте на апстрактни количнички тип што производи операције

[4]Све функције (укључујући и апстрактну функцију $\lceil\_\rceil^{C2}$) у HOL су тоталне. Ипак, све леме о тој функцији које су доказане, садрже један додатни услов, а то је да њихов аргумент није $(0,0)$. Зато, не постоји разлог да резонујемо о вредности $\lceil(0,0)\rceil^{C2}$ и може се сматрати као лоше дефинисана вредност.

означене са $\texttt{uminus}_{hc}$, $*_{hc}$, и $\texttt{recip}_{hc}$. Одузимање (означено са $-_{hc}$) је дефинисано коришћењем $+_{hc}$ и $\texttt{uminus}_{hc}$, а дељење (означено са $:_{hc}$) се дефинише коришћењем $*_{hc}$ и $\texttt{recip}_{hc}$. Као и у случају сабирања, доказано је да све ове операције одговарају обичним операцијама коначне комплексне равни (нпр. **lemma** "$\texttt{uminus}_{hc}$ (of_complex z) = of_complex $(-z)$"). Следеће леме показују понашање ових операција када се у њима појављује и тачка бесконачно (приметимо да изрази $0_{hc} *_{hc} \infty_{hc}$, $\infty_{hc} *_{hc} 0_{hc}$, $0_{hc} :_{hc} 0_{hc}$, и $\infty_{hc} :_{hc} \infty_{hc}$ су лоше дефинисани).

lemma "$\texttt{uminus}_{hc}\ \infty_{hc} = \infty_{hc}$"
lemma "$\texttt{recip}_{hc}\ \infty_{hc} = 0_{hc}$" "$\texttt{recip}_{hc}\ 0_{hc} = \infty_{hc}$"
lemma "$z \neq 0_{hc} \Longrightarrow z *_{hc} \infty_{hc} = \infty_{hc} \ \wedge \ \infty_{hc} *_{hc} z = \infty_{hc}$"
lemma "$z \neq 0_{hc} \Longrightarrow z :_{hc} \infty_{hc} = 0_{hc}$"
lemma "$z \neq \infty_{hc} \Longrightarrow \infty_{hc} :_{hc} z = \infty_{hc}$"

Такође, проширен је и комплексни коњугат (на репрезентативном типу (z_1, z_2) је мапирано на $(\overline{z_1}, \overline{z_2})$), што даје операцију $\texttt{cnj}_{hc}$. Веома важна операција у комплексној геометрији је *инверзија у односу на јединични круг*:

definition $\texttt{inversion}_{hc}$ :: "$\texttt{complex}_{hc} \Rightarrow \texttt{complex}_{hc}$" **where**
"$\texttt{inversion}_{hc} = \texttt{cnj}_{hc} \circ \texttt{recip}_{hc}$"

Основне особине инверзије се лако доказују.

lemma "$\texttt{inversion}_{hc} \circ \texttt{inversion}_{hc} = id$"
lemma "$\texttt{inversion}_{hc}\ 0_{hc} = \infty_{hc}$" "$\texttt{inversion}_{hc}\ \infty_{hc} = 0_{hc}$"

Размера и дворазмера.

Размера и дворазмера су веома важни појмови у пројективној геометрији и у проширеној комплексној равни (дворазмера се карактерише као инваријанта Мебијусових трансформација – основних трансформација у $\overline{\mathbb{C}}$, и могуће је дефинисати праве коришћењем размера и круга коришћењем дворазмере).

Размера тачака z, v и w се обично дефинише као $\frac{z-v}{z-w}$. Наша дефиниција уводи хомогене координате.

definition ratio_rep **where** "ratio_rep z v w =
(**let** $(z_1,\ z_2) = \lfloor z \rfloor_{C2}$; $(v_1,\ v_2) = \lfloor v \rfloor_{C2}$; $(w_1,\ w_2) = \lfloor w \rfloor_{C2}$

```
    in ⌈((z₁ * v₂ − v₁ * z₂) * w₂, (z₁ * w₂ − w₁ * z₂) * v₂)⌉^C2)"
lift_definition ratio ::
  "complex_hc ⇒ complex_hc ⇒ complex_hc ⇒ complex_hc" is ratio_rep
```

Приметимо да је ово добро дефинисано у свим случајевима осим када важи $z = w = v$ или $z = v = \infty_{hc}$ или $z = w = \infty_{hc}$ или $v = w = \infty_{hc}$ (ипак, у доказима код подизања на количнички тип ови лоше дефинисани случајеви такође морају бити анализирани). Додатно, оригинална размера разлика је дефинисана у свим случајевима осим када $z = w = v$ или $z = \infty_{hc}$ или $v = w = \infty_{hc}$, тако да наша дефиниција у хомогеним координатама природно проширује оригиналну дефиницију. Следеће леме показују понашање размере у свим добро дефинисаним случајевима (одговара оригиналној размери разлика кад год је она дефинисана).

```
lemma "⟦z ≠ v ∨ z ≠ w;  z ≠ ∞_hc;  v ≠ ∞_hc ∨ w ≠ ∞_hc⟧ ⟹
  ratio z v w = (z −_hc v) :_hc (z −_hc w)"
lemma "⟦v ≠ ∞_hc;  w ≠ ∞_hc⟧ ⟹ ratio ∞_hc v w = 1_hc"
lemma "⟦z ≠ ∞_hc;  w ≠ ∞_hc⟧ ⟹ ratio z ∞_hc w = ∞_hc"
lemma "⟦z ≠ ∞_hc;  v ≠ ∞_hc⟧ ⟹ ratio z v ∞_hc = 0_hc"
```

Последње две леме су последице прве леме. Такође, приметимо да размера не може бити дефинисана на природан начин у случају када су барем две тачке бесконачно (тако да функција размере остане непрекидна по свим својим параметрима).

Дворазмера је дефинисана над 4 тачке (z, u, v, w), обично као $\frac{(z-u)(v-w)}{(z-w)(v-u)}$. Поново, ми је дефинишемо користећи хомогене координате.

```
definition cross_ratio_rep where "cross_ratio_rep z u v w =
  (let (z₁, z₂) = ⌊z⌋_C2;  (u₁, u₂) = ⌊u⌋_C2;
       (v₁, v₂) = ⌊v⌋_C2;  (w₁, w₂) = ⌊w⌋_C2 in
  ⌈(z₁ * u₂ − u₁ * z₂) * (v₁ * w₂ − w₁ * v₂), (z₁ * w₂ − w₁ * z₂) * (v₁ * u₂ − u₁ * v₂))⌉^C2"
lift_definition cross_ratio :: "complex_hc ⇒ complex_hc ⇒
  complex_hc ⇒ complex_hc ⇒ complex_hc" is cross_ratio_rep
```

Ово је добро дефинисано у свим случајевима осим када $z = u = w$ или $z = v = w$ или $z = u = v$ или $u = v = w$ (приметимо да су бесконачне вредности за z, u, v или w дозвољене, што није случај у оригиналној формулацији разломка). Нека основна својства дворазмере су дата следећим лемама.

lemma "$[\![(z \neq u \wedge v \neq w) \vee (z \neq w \wedge u \neq v); z \neq \infty_{hc}; u \neq \infty_{hc}; v \neq \infty_{hc}\ w \neq \infty_{hc}]\!]$
$\Longrightarrow$ cross_ratio $z\ u\ v\ w$ = $((z -_{hc} u) *_{hc} (v -_{hc}) :_{hc} ((z -_{hc} w) *_{hc} (v -_{hc} u))$"
lemma "cross_ratio $z\ 0_{hc}\ 1_{hc}\ \infty_{hc}$ = z"
lemma "$[\![\ z_1 \neq z_2; z_1 \neq z_3\ ]\!] \Longrightarrow$ cross_ratio $z_1\ z_1\ z_2\ z_3$ = 0_{hc}"
lemma "$[\![\ z_2 \neq z_1; z_2 \neq z_3\ ]\!] \Longrightarrow$ cross_ratio $z_2\ z_1\ z_2\ z_3$ = 1_{hc}"
lemma "$[\![\ z_3 \neq z_1; z_3 \neq z_2\ ]\!] \Longrightarrow$ cross_ratio $z_3\ z_1\ z_2\ z_3$ = ∞_{hc}"

5.4 Риманова сфера и стереографска пројекција

Проширена комплексна раван се може идентификовати са Римановом (јединичном) сфером Σ коришћењем стереографске пројекције [130, 148]. Сфера се пројектује из свог северног пола N на xOy раван (коју означавамо са $\mathbb{C}$). Ова пројекција успоставља бијективно пресликавање sp између $\Sigma \setminus N$ и коначне комплексне равни $\mathbb{C}$. Тачка бесконачно је дефинисана као слика од N.

У *Isabelle/HOL* систему, сфера Σ је дефинисана као нови тип.

typedef riemann_sphere = "{(x, y, z)::R3_vec. $x^2 + y^2 + z^2 = 1$}"

Као и раније, ово дефинише функцију Rep_riemann_sphere (која је означена са $\lfloor\_\rfloor_{R3}$) и функцију Abs_riemann_sphere (која је означена са $\lceil\_\rceil^{R3}$) која повезује тачке апстрактног типа (riemann_sphere) и тачке репрезантативног типа (тројке реалних бројева). Стереографска пројекција се уводи на следећи начин:

definition stereographic_rep :: "riemann_sphere $\Rightarrow$ C2_vec$_{\neq 0}$" **where**
"stereographic_rep M =
(let (x, y, z) = $\lfloor M \rfloor_{R3}$
in if (x, y, z) $\neq$ (0, 0, 1) then $\lceil (x + i * y, 1 - z) \rceil^{C2}$
else $\lceil (1, 0) \rceil^{C2}$)"
lift_definition stereographic :: "riemann_sphere $\Rightarrow$ complex$_{hc}$" **is**
stereographic_rep

За све тачке, ово је добро дефинисано (вектор $(x + i * y, 1 - z)$ је ненула јер $(x, y, z) \neq (0, 0, 1)$, и $(1, 0)$ је очито ненула).

Инверзна стереографска пројекција се дефинише на следећи начин.

```
definition inv_stereographic_rep :: "C2_vec≠0 ⇒ riemann_sphere"
where
   "inv_stereographic_rep z =
       (let (z1, z2) = ⌊z⌋C2
         in if z2 = 0 then ⌈(0,0,1)⌉R3
            else let z = z1/z2; XY = (2*z)/cor (1+|z|²);
            Z = (|z|²-1)/(1+|z|²)
                  in ⌈(Re XY, Im XY, Z)⌉R3)"
lift_definition inv_stereographic :: "complexhc ⇒ riemann_sphere" is
   inv_stereographic_rep
```

За све тачке, ово је добро дефинисано (сума квадрата три координате је 1 у оба случаја, па се може применити функција `Abs_riemann_sphere`).

Веза између две функције је дата следећим лемама.

```
lemma "stereographic ∘ inv_stereographic = id"
lemma "inv_stereographic ∘ stereographic = id"
lemma "bij stereographic" "bij inv_stereographic"
```

Докази нису тешки, али захтевају формализацију врло незгодних израчунавања.

Тетивно растојање. Риманова сфера може бити метрички простор. Најчешћи начин да се уведе метрички простор је коришћењем *тетивне, Риманове метрике* – растојање између две тачке на сфери је дужина тетиве која их спаја.

```
definition distrs :: "riemann_sphere ⇒ riemann_sphere ⇒ real" where
  "distrs M1 M2 = (let (x1, y1, z1) = ⌊M1⌋R3; (x1, y1, z1) = ⌊M2⌋R3
                    in norm (x1 - x2, y1 - y2, z1 - z2))"
```

Други приступ је да се узме *лучно растојање*, односно узети дужину одговарајућег кружног лука, односно одговарајући централни угао у радијанима.

Функција `norm` је уграђена функција и у овом случају она рачуна еуклидску векторску норму. Коришћењем (сада већ познате) чињенице да је $\mathbb{R}^3$

метрички простор (са функцијом растојања $\lambda\ x\ y.\ norm(x - y)$), није било тешко доказати да је тип `riemann_sphere` опремљен са $\texttt{dist}_{rs}$ метрички простор, тј. да је он инстанца локала `metric_space`. Иако је дефинисана на сфери, тетивна метрика има своју репрезентацију и у равни.

```
lemma assumes
 "stereographic M₁ = of_complex m₁"
 "stereographic M₂ = of_complex m₂"
 shows "dist_rs M₁ M₂ = 2*|m₁-m₂| / (sqrt (1+|m₁|²)*sqrt (1+|m₂|²))"
lemma assumes
 "stereographic M₁ = ∞_hs"
 "stereographic M₂ = of_complex m"
 shows "dist_rs M₁ M₂ = 2 / sqrt (1+|m|²)"
lemma assumes
 "stereographic M₁ = of_complex m"
 "stereographic M₂ = ∞_hs"
 shows "dist_rs M₁ M₂ = 2 / sqrt (1+|m|²)"
lemma assumes "stereographic M₁ = ∞_hs" "stereographic M₂ = ∞_hs"
 shows "dist_rs M₁ M₂ = 0"
```

Ове леме праве разлику између коначних и бесконачних тачака, али се ова анализа случаја може избећи коришћењем хомогених координата.

```
definition "⟪z, w⟫ = (vec_cnj ⌊z⌋_C2) *_vv (⌊w⌋_C2)"
definition "⟪z⟫ = sqrt (Re ⟪z, z⟫)"
definition "dist_hc_rep = 2*sqrt(1 - |⟪z, w⟫|² / (⟪z⟫² * ⟪w⟫²))"
lift_definition dist_hc :: "complex_hc ⇒ complex_hc ⇒ real" is
  dist_hc_rep
lemma "dist_rs M₁ M₂ = dist_hc (stereographic M₁) (stereographic M₂)"
```

Понекад се ова форма зове Фубини–Стади (енг. *Fubini–Study*) метрика.

Тип $\texttt{complex}_{hc}$ опремљен са $\texttt{dist}_{hc}$ метриком је такође инстанца локала `metric_space`. Ово тривијално следи из последње леме која је повезује са метричким простором на Римановој сфери. Постоје и директни докази ове чињенице (нпр. Хил (енг. *Hille*) [82] даје директан доказ захваљујући Какутани (енг. *Kakutani*), али доказ је некомплетан јер занемарује могућност да

једна тачка буде бесконачно), а ми смо и те директне доказе формализовали[5]. Испоставило се да је нека својства лакше доказати на Римановој сфери коришћењем функције $\texttt{dist}_{rs}$ (нпр. неједнакост троугла), али нека својства је било лакше доказати у пројекцији коришћењем функције $\texttt{dist}_{hc}$ (нпр. да је метрички простор савршен, тј. да нема изолованих тачака), што показује значај постојања различитих модела за исти концепт.

Коришћењем тетивне метрике у проширеној комплексној равни, и еуклидске метрике на сфери у $\mathbb{R}^3$, доказано је да су стереографска пројекција и инверзна стереографска пројекција непрекидне.

```
lemma "continuous_on UNIV stereographic"
      "continuous_on UNIV inv_stereographic"
```

Приметимо да је у претходној леми, метрика имплицитна (у систему *Isabelle/HOL* претпоставља се да коришћена метрика је управо она метрика која је коришћена да се докаже да је дати тип инстанца локала `metric_space`).

5.5 Мебијусове трансформације

Мебијусове трансформације (које се још називају и холоморфна пресликавања, линеарна фракциона трансформација или билинеарна пресликавања) су основне трансформације проширене комплексне равни. У нашој формализацији оне су уведене алгебарски. Свака трансформација је представљена регуларном (несингуларном, недегенерисаном) 2×2 матрицом која линеарно делује на хомогене координате. Како пропорционалне хомогене координате представљају исту тачку у $\overline{\mathbb{C}}$, тако и пропорционалне матрице представљају исту Мебијусову трансформацију. Поново, формализација се састоји из три корака коришћењем *lifting/transfer* пакета. Прво, уводи се тип регуларних матрица.

```
typedef C2_mat_reg = "{M :: C2_mat. mat_det M ≠ 0}"
```

[5]Наша формализација је започета без анализирања Риманове сфере, тако да смо у почетку једино и могли користити директне доказе, али у неком тренутку увели смо појам Риманове сфере и то је помогло да се многи докази упросте, укључујући и овај.

Функција репрезентације `Rep_C2_mat_reg` ће бити означена са $\lfloor\_\rfloor_M$, а апстрактна функција `Abs_C2_mat_reg` ће бити означена са $\lceil\_\rceil^M$. Регуларне матрице формирају групу у односу на множење и она се често назива *генерална линеарна група* и означава се са $GL(2,\mathbb{C})$. У неким случајевима се разматра само њена подгрупа, *специјална линеарна група*, означена са $SL(2,\mathbb{C})$, која садржи само оне матрице чија је детерминанта једнака 1.

Мебијусова група. Кажемо да су две регуларне матрице еквивалентне ако су њихове репрезентације пропорционалне.

```
definition ≈M :: "C2_mat_reg ⇒ C2_mat_reg ⇒ bool" where
  "M1 ≈M M2 ⟷ (∃ (k::complex). k ≠ 0 ∧ ⌊M2⌋M = k *sm ⌊M1⌋M)"
```

Лако се доказује да је ова релација заправо релација еквиваленције. Елементи Мебијусове групе се уводе као класа еквиваленције над овом релацијом.

```
quotient_type mobius = C2_mat_reg / ≈M
```

Понекад ћемо користити помоћни конструктор `mk_mobius` који враћа елемент Мебијусове групе (класу еквиваленције) за дата 4 комплексна параметра (што има смисла само када је одговарајућа матрица регуларна).

Мебијусови елементи формирају групу над композицијом. Ова група се назива *пројективна генерална линеарна група* и означена је са $PGL(2,\mathbb{C})$. Поново, могу се разматрати само они елементи *специјалне пројективне групе* $SGL(2,\mathbb{C})$ чија је детерминанта једнака 1. Композиција Мебијусових елемената се постиже множењем матрица које их репрезентују.

```
definition mobius_comp_rep :: "C2_mat_reg ⇒ C2_mat_reg ⇒ C2_mat_reg"
   where "moebius_comp_rep M1 M2 = ⌈⌊M1⌋M *mm ⌊M2⌋M⌉M"
lift_definition mobius_comp :: "mobius ⇒ mobius ⇒ mobius" is
   mobius_comp_rep
```

Слично, инверзна Мебијусова трансформација се добија инверзијом матрице која је представља.

```
definition mobius_inv_rep :: "C2_mat_reg ⇒ C2_mat_reg" where
   "mobius_inv_rep M = ⌈mat_inv ⌊M⌋M⌉M"
lift_definition mobius_inv :: "mobius ⇒ mobius" is "mobius_inv_rep"
```

Коначно, Мебијусова трансформација која је идентитет је представљена јединичном матрицом.

```
definition mobius_id_rep :: "C2_mat_reg" where
   "mobius_id_rep = ⌈eye⌉^M"
lift_definition mobius_id :: "mobius" is mobius_id_rep
```

Све ове дефиниције увек уводе добро дефинисане објекте (јер је производ регуларних матрица регуларна матрица, а инверз регуларне матрице је такође регуларна матрица). Обавезни услови да би се дефиниција могла подићи (нпр. $M_1 \approx_M M_2 \implies$ `mobius_inv_rep` $M_1 \approx_M$ `mobius_inv_rep` M_2) се лако доказују. Онда, доказује се да је тип `mobius` заједно са овим операцијама инстанца локала `group_add` који је већ уграђен у систем *Isabelle/HOL*. Зато, ми ћемо понекад означавати `mobius_comp` са $+$, `mobius_inv` са унарним $-$, и `mobius_id` са 0.

Дејство Мебијусове групе. Мебијусове трансформације су дефинисане као дејство Мебијусове групе на тачке проширене комплексне равни (које су дате у хомогеним координатама).

```
definition mobius_pt_rep :: "C2_mat_reg ⇒ C2_vec_{≠0} ⇒ C2_vec_{≠0}"
  where "moebius_pt_rep M z = ⌈⌊M⌋_M *_mv ⌊z⌋_C2⌉^C2"
lift_definition mobius_pt :: "mobius ⇒ complex_hc ⇒ complex_hc" is
  mobius_pt_rep
```

Како производ регуларне матрице и ненула вектора је увек ненула вектор, резултат је увек добро дефинисан. Подизање дефиниција генерише обавезан услов $[\![M \approx_M M';\ z \approx_{C2} z']\!] \implies$ `mobius_pt_rep` $M\ z \approx_{C2}$ `mobius_pt_rep` $M'\ z'$ који се прилично лако доказује.

Када се узима у обзир дејство групе на проширену комплексну раван, онда се може видети да операције групе заиста одговарају композицији пресликавања, инверзном пресликавању и идентичном пресликавању.

```
lemma "mobius_pt (mobius_comp M_1 M_2) =
                 (mobius_pt M_1) ∘ (mobius_pt M_2)"
lemma "mobius_pt (mobius_inv M) = inv (mobius_pt M)"
lemma "mobius_pt (mobius_id) = id"
```

Дејство је транзитивно (јер је увек бијективно пресликавање).

```
lemma "bij (mobius_pt M)"
```

У класичној литератури Мебијусове трансформације се обично представљају у форми $\frac{az+b}{cz+d}$, и наредна лема заиста оправдава и овакав запис (али у њој разликујемо специјалан случај када је z тачка бесконачно).

```
lemma assumes "mat_det (a, b, c, d) ≠ 0"
      shows "moebius_pt (mk_mobius a b c d) z =
                  (if z ≠ ∞hc then
                        ((of_complex a) *hc z +hc (of_complex b)) :hc
                        ((of_complex c) *hc z +hc (of_complex d))
                  else (of_complex a) :hc (of_complex c))"
```

Произвољна трансформација у $\overline{\mathbb{C}}$ ће бити звана Мебијусовом трансформацијом акко је она дејство неког елемента Мебијусове групе.

```
definition is_mobius :: "(complexhc ⇒ complexhc) ⇒ bool" where
  "is_mobius f ⟷ (∃ M. f = mobius_pt M)"
```

Приметимо да већина до сада изнетих резултата зависи од чињенице да је матрица репрезентације Мебијусове трансформације регуларна – у супротном, дејство би било дегенерисано и целу раван $\overline{\mathbb{C}}$ би сликало у једну тачку.

Неке специјалне Мебијусове трансформације.

Многе трансформације са којима се сусрећемо у геометрији су заправо специјална врста Мебијусових трансформација. Веома важна подгрупа је група *еуклидских сличности* (које се још називају и *интегралне трансформације*). Оне су одређене са два комплексна параметра (и представљају Мебијусову трансформацију када први од та два параметра није нула).

```
definition similarity :: "complex ⇒ complex ⇒ mobius" where
  "similarity a b = mk_mobius a b 0 1"
```

Сличности формирају групу (која се понекад назива и *параболичка група*).

```
lemma "⟦a ≠ 0; c ≠ 0⟧ ⟹
  mobius_comp (similarity a b) (similarity c d) =
```

```
  similarity (a * c) (a * d + b)"
lemma "a ≠ 0 ⟹
  mobius_inv (similarity a b) = similarity (1/a) (-b/a)"
lemma "id_mobius = similarity 1 0"
```

Њихово дејство је линеарна трансформација у $\mathbb{C}$, а свака линеарна трансформација у $\mathbb{C}$ која није константна је дејство елемента групе еуклидских сличности.

```
lemma "a ≠ 0 ⟹ mobius_pt (similarity a b) =
                (λ z. (of_complex a) *hc z +hc (of_complex b))"
```

Еуклидске сличности су једини елементи Мебијусове групе такви да је тачка ∞_{hc} фиксна тачка.

```
lemma "mobius_pt M ∞hc = ∞hc ⟷
    (∃ a b. a ≠ 0 ∧ M = similarity a b)"
```

Ако су и тачка ∞_{hc} и тачка 0_{hc} фиксне, онда је то сличност са коефицијентима $a \neq 0$ и $b = 0$, а дејство је облика $\lambda\ z.\ $ `(of_complex a)` $*_{hc}\ z$.

```
lemma "mobius_pt M ∞hc = ∞hc ∧ mobius_pt M 0hc = 0hc ⟷
  (∃ a. a ≠ 0 ∧ M = similarity a 0)"
```

Еуклидске сличности укључују транслацију, ротацију и хомотетију и свака еуклидска сличност се може добити као композиција ове три врсте пресликавања.

```
definition "translation v = similarity 1 v"
definition "rotation φ    = similarity (cis φ) 0"
definition "dilatation k  = similarity (cor k) 0"
lemma "a ≠ 0 ⟹ similarity a b =
   (translation b) + (rotation (arg a)) + (dilatation |a|)"
```

Реципрочна вредност $(1_{hc} :_{hc} z)$ је такође Мебијусова трансформација.

```
definition "reciprocation = mk_mobius (1, 0, 0, 1)"
lemma "recip_hc = mobius_pt reciprocation"
```

Са друге стране, инверзија није Мебијусова трансформација (то је основни пример такозваних анти-Мебијусових трансформација, или антихоломорфне функције).

Веома важна чињеница је да се свака Мебијусова трансформација може добити композицијом еуклидских сличности и реципрочне функције. Један од начина како се ово може постићи дат је следећом лемом (када је $c = 0$ је случај еуклидских сличности и ово је раније већ анализирано).

```
lemma assumes "c ≠ 0" and "a * d - b * c ≠ 0"
shows "mk_mobius a b c d =
       translation (a/c) + rotation_dilatation ((b*c - a*d)/(c*c)) +
       reciprocal + translation (d/c)"
```

Декомпозиција је коришћена у многим доказима. Наиме, да бисмо доказали да свака Мебијусова трансформација има неко својство, довољно је доказати да реципрочна функција и еуклидске сличности задовољавају то својство и да композиција чува то својство (обично, најтеже је доказати у случају реципрочне функције, а остала два корака буду углавном много једноставнија).

```
lemma assumes "⋀ v. P (translation v)" "⋀ α. P (rotation α)"
  "⋀ k. P (dilatation k)" "P (reciprocation)"
  "⋀ M₁ M₂. ⟦ P M₁; P M₂ ⟧ ⟹ P (M₁ + M₂)"
  shows "P M"
```

Дворазмера као Мебијусова трансформација

За било које три фиксне тачке z_1, z_2 и z_3, `cross_ratio` z z_1 z_2 z_3 се може посматрати као функција једне промењиве z. Следећа лема гарантује да је ова функција Мебијусова трансформација и да према особина дворазмере она слика z_1 у 0_{hc}, z_2 у 1_{hc} и z_3 у ∞_{hc}.

```
lemma "⟦ z₁ ≠ z₂; z₁ ≠ z₃; z₂ ≠ z₃ ⟧ ⟹
  is_mobius (λ z. cross_ratio z z₁ z₂ z₃)"
```

Доказавши ово тврђење, дворазмера се може користити да се докаже да постоји Мебијусова трансформација која слика било које три различите тачке

редом у 0_{hc}, 1_{hc} и ∞_{hc}. Како Мебијусове трансформације чине групу, једноставна последица овога је да постоји Мебијусова трансформација која слика било које три различите тачке у било које три различите тачке.

```
lemma "⟦ z1 ≠ z2; z1 ≠ z3; z2 ≠ z3 ⟧ ⟹ (∃ M. mobius_pt M z1 = 0hc ∧
  mobius_pt M z2 = 1hc ∧ mobius_pt M z3 = ∞hc)"
```

Следећа лема има веома важну примену у даљем развоју теорије јер омогућава закључивање „без губитка на општости (бгно)" [78]. Наиме, ако Мебијусова трансформација чува неко својство, онда уместо три произвољне тачке може се посматрати само случај специјалних тачака 0_{hc}, 1_{hc}, и ∞_{hc}.

```
lemma assumes "P 0hc 1hc ∞hc" "z1 ≠ z2" "z1 ≠ z3" "z2 ≠ z3"
  "⋀ M u v w. P u v w ⟹
            P (mobius_pt M u) (mobius_pt M b) (mobius_pt M c)"
  shows "P z1 z2 z3"
```

Једна од првих примена „бгно" резоновања за Мебијусове трансформације је у анализи фиксних тачака Мебијусових трансформација. Лако се доказује да једино идентично пресликавање има фиксне тачке 0_{hc}, 1_{hc}, и ∞_{hc}. Такође важи да ако Мебијусова трансформација M има три различите фиксне тачке, онда је она идентитет, али директан доказ овога се заснива на чињеници да 2×2 матрица има највише два независна сопствена вектора, а овакво закључивање се лако може избећи коришћењем „бгно" резоновања (како било које три тачке можемо сликати редом у 0_{hc}, 1_{hc}, и ∞_{hc} неким пресликавањем M', а онда пресликавање $M' + M - M'$ има ове три тачке фиксне па мора бити једнако 0).

```
lemma "⟦ mobius_pt M 0hs = 0hs; mobius_pt M 1hs = 1hs;
  mobius_pt M ∞hs = ∞hs ⟧ ⟹ M = id_mobius"
lemma "⟦ mobius_pt M z1 = z1; mobius_pt M z2 = z2;
  mobius_pt M z3 = z3; z1 ≠ z2; z1 ≠ z3; z2 ≠ z3 ⟧ ⟹ M = id_mobius"
```

Последица овога је да постоји јединствена Мебијусова трансформација која слика три различите тачке у друге три различите тачке (већ је доказано да такво пресликавање постоји, а ако би постојала два таква пресликавања онда би њихова разлика морала имати три фиксне тачке, што значи да би била идентитет).

lemma "$[\![z_1 \neq z_2;\ z_1 \neq z_3;\ z_2 \neq z_3;\ w_1 \neq w_2;\ w_1 \neq w_3;\ w_2 \neq w_3]\!] \Longrightarrow \exists!\ M.$
 mobius_pt M z_1 = w_1 $\wedge$ mobius_pt M z_2 = w_2 $\wedge$ mobius_pt M z_3 = w_3"

Мебијусове трансформације чувају дворазмеру. Поново, директан доказ би био компликован, па је формализован елегантан индиректни доказ (у основи, разлика $\lambda z.$ cross_ratio z z_1 z_2 z_3 и M слика (M z_1) у 0_{hc}, (M z_2) у 1_{hc}, и (M z_3) у ∞_{hc}, па зато мора бити једнака $\lambda z.$ cross_ratio z (M z_1) (M z_2) (M z_3), и тврђење следи замењујући (M z) са z).

lemma "$[\![z_1 \neq z_2;\ z_1 \neq z_3;\ z_2 \neq z_3]\!] \Longrightarrow$
    cross_ratio z z_1 z_2 z_3 =
    cross_ratio (mobius_pt M z) (mobius_pt M z_1)
                (mobius_pt M z_2) (mobius_pt M z_3)"

5.6 Кругоправа

Веома важно својство проширене комплексне равни је могућност да праве и кругове посматрамо униформно. Основни објекат је *уопштен круг* или скраћено *кругоправа*. У нашој формализацији ми смо пратили приступ који је описао Швердфегер [148] и представили смо кругоправе хермитским, ненула 2×2 матрицама. У оригиналној формулацији, матрица $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ одговара једначини $A * z *$ cnj $z + B *$ cnj $z + C * z + D = 0$, где је $C =$ cnj B и A и D су реални (јер је матрица хермитска). Кључно је да ова једначина представља праву када је $A = 0$, а иначе круг.

Поново, наша формализација се састоји из три корака. Прво, уведен је тип хермитских, ненула матрица.

definition is_C2_mat_herm :: "C2_mat ⇒ bool" **where**
  "is_C2_mat_herm H $\longleftrightarrow$ hermitean H $\wedge$ H $\neq$ mat_zero"
typedef C2_mat_herm = "{H :: C2_mat. is_C2_mat_herm H}"

Функција репрезентације Rep_C2_mat_herm ће бити означена са $\lfloor \_ \rfloor_H$, а апстрактна функција Abs_C2_mat_herm ће бити означена са $\lceil \_ \rceil^H$. Имајући на уму интерпретацију у форми једначине, јасно је да би поново пропорционалне матрице требало сматрати еквивалентним. Овог пута, фактор пропорционалности матрица је реалан ненула број.

```
definition ≈cm :: "C2_mat_herm ⇒ C2_mat_herm ⇒ bool" where
  "H1 ≈cm H2 ⟷ (∃ (k::real). k ≠ 0 ∧ ⌊H2⌋H = cor k *sm ⌊H1⌋H)"
```

Лако се доказује да је ово релација еквиваленције, а кругоправе се дефинишу коришћењем количничке конструкције као класа еквиваленције.

```
quotient_type circline = C2_mat_herm / ≈cm
```

Помоћни конструктор `mk_circline` даје кругоправу (класу еквиваленције) за дата четири комплексна броја A, B, C и D (под претпоставком да они формирају хермитску, ненула матрицу).

Свака кругоправа одређује одговарајући скуп тачака. Поново, опис који је дат у хомогеним координатама је нешто бољи него оригинални опис који је дат за обичне комплексне бројеве. Тачка са хомогеним координатама (z_1, z_2) ће припадати скупу тачака кругоправе акко $A * z_1 * \texttt{cnj}\ z_1 + B * \texttt{cnj}\ z_1 * z_2 + C * z_1 * \texttt{cnj}\ z_2 + D * z_2 * \texttt{cnj}\ z_2 = 0$. Приметимо да је ово квадратна форма која је одређена вектором хомогених координата и хермитском матрицом. Зато, скуп тачака на датој кругоправој се формализује на следећи начин (поред дефиниције кругоправе на овом месту дајемо и дефиниције билинеарне и квадратне форме које су уведене у нашој основној теорији линеарне алгебре).

```
definition "bilinear_form H z1 z2 = (vec_cnj z1) *vm H *vv z2"
definition "quad_form H z = bilinear_form H z z"
definition on_circline_rep :: "C2_mat_herm ⇒ C2_vec≠0 ⇒ bool" where
  "on_circline_rep H z ⟷ quad_form ⌊H⌋H ⌊z⌋C2 = 0"
lift_definition on_circline :: "circline ⇒ complexhc ⇒ bool" is
  on_circline_rep
definition circline_set :: "complexhc set" where
  "circline_set H = {z. on_circline H z}"
```

Подизање дефиниције `on_circline` ствара услове $[\![H_1 \approx_{cm} H_2;\ z_1 \approx_{C2} z_2]\!] \Longrightarrow$ `on_circline_rep` $H_1\ z_1 \longleftrightarrow$ `on_circline_rep` $H_2\ z_2$ који се лако доказују.

Неке специјалне кругоправе. Међу свим кругоправама најзначајније су јединични круг, x-оса и имагинарни јединични круг.

```
definition "unit_circle_rep = ⌈(1, 0, 0, −1)⌉H"
lift_definition unit_circle :: "circline" is unit_circle_rep
```

```
definition "x_axis_rep = ⌈(0, i, −i, 0)⌉^H"
lift_definition x_axis :: "circline" is x_axis_rep
definition "imag_unit_circle_rep = ⌈(1, 0, 0, 1)⌉^H"
lift_definition imag_unit_circle :: "circline" is
    imag_unit_circle_rep
```

Лако се доказују нека основна својства ових кругоправих. На пример:

```
lemma "0_hc ∈ circline_set x_axis" "1_hc ∈ circline_set x_axis"
      "∞_hc ∈ circline_set x_axis"
```

Повезаност са правама и круговима у обичној еуклидској равни.

У проширеној комплексној равни не постоји разлика између појма праве и појма круга. Ипак, праве могу бити дефинисане као оне кругоправе код којих матрице имају коефицијент $A = 0$, или, еквивалентно као оне кругоправе које садрже тачку ∞_{hc}.

```
definition is_line_rep where
  "is_line_rep H ⟷ (let (A, B, C, D) = ⌊H⌋_H in A = 0)"
lift_definition is_line :: "circline ⇒ bool" is is_line_rep
definition is_circle_rep where
  "is_circle_rep H ⟷ (let (A, B, C, D) = ⌊H⌋_H in A ≠ 0)"
lift_definition is_circle :: "circline ⇒ bool" is is_circle_rep
lemma "is_line H ⟷ ¬ is_circle H" "is_line H ∨ is_circle H"
lemma "is_line H ⟷ ∞_hc ∈ circline_set H"
      "is_circle H ⟷ ∞_hc ∉ circline_set H"
```

Сваки еуклидски круг и еуклидкса права (у обичној комплексној равни, коришћењем стандардне, еуклидске метрике) може бити представљена коришћењем кругоправе.

```
definition mk_circle_rep μ r = ⌈(1, −μ, −cnj μ, |μ|^2 − (cor r)^2)⌉^H
lift_definition mk_circle :: "complex ⇒ real ⇒ circline" is
  mk_circle_rep
lemma "r ≥ 0 ⟹ circline_set (mk_circle μ r) =
                    of_complex ` {z. |z − μ| = r}"
definition mk_line_rep where "mk_line_rep z_1 z_2 =
```

```
  (let B = i * (z2 - z1) in ⌈(0, B, cnj B, -(B * cnj z1 + cnj B * z1)⌉^H)"
lift_definition mk_line :: "complex ⇒ complex ⇒ circline" is
  mk_line_rep
lemma "z1 ≠ z2 ⟹ circline_set (mk_line z1 z2) - {∞hc} =
                      of_complex ` {z. collinear z1 z2 z}"
```

Супротно такође важи, скуп тачака који су одређени кругоправом је увек или еуклидски круг или еуклидска права. Следећа функција одређује параметре круга или параметре праве (центар и полупречник у случају круга или две различите тачке у случају праве) за дату кругоправу.

```
definition euclidean_circle_rep where "euclidean_circle_rep H =
  (let (A, B, C, D) = ⌊H⌋_H
    in (-B/A, sqrt(Re ((B * C - A * D)/(A * A)))))"
lift_definition euclidean_circle :: "circline ⇒ complex × real" is
  euclidean_circle_rep
definition euclidean_line_rep where "euclidean_line_rep H =
  (let (A, B, C, D) = ⌊H⌋_H;
       z1 = -(D * B)/(2 * B * C);
       z2 = z1 + i * sgn (if arg B > 0 then -B else B)
    in (z1, z2))"
lift_definition euclidean_line :: "circline ⇒ complex × complex" is
  euclidean_line_rep
```

Приметимо да је нормални вектор праве вектор који је нормалан на вектор који је одређен координатним почетком и комплексним бројем B (коефицијент матрице), односно $z_2 = z_1 + i * B$. Да бисмо могли да подигнемо дефиницију (тако да су добијене тачке исте за сваку матрицу која репрезентује исту кругоправу) у дефиницији друге тачке вектор B је нормализован. Ово даје нешто већи израз од израза $z_2 = z_1 + i * B$.

Додатно, кардиналност скупа тачака кругоправе зависи од знака израза $\mathtt{Re}((B * C - A * D)/(A * A))$. Зато, кругоправе могу бити класификоване у три категорије у зависности од знака детерминанте (која је увек реалан број, јер је матрица хермитска).

```
definition circline_type_rep where
  "circline_type_rep H = sgn (Re (mat_det (⌊H⌋_H)))"
```

```
lift_definition circline_type :: "circline ⇒ real" is
  circline_type_rep
```

Обавезан услов $H \approx_{cm} H' \Longrightarrow$ `circline_type_rep` H = `circline_type_rep` H' се лако доказује, јер `Re (mat_det (`k $*_{sm}$ H`)) = (Re` k`)`^{2} `* Re (mat_det` H`)` важи за све хермитске матрице H и за све k са имагинарним делом 0.

Сада постаје јасно да је скуп тачака на датој кругоправој празан акко је тип кругоправе позитиван (ове кругоправе се зову *имаѓинарне кругоūраве*), да садржи само једну тачку акко је тип кругоправе једнак нули (*ūачка кругоūраве*) и да је бесконачан акко је тип негативан (*реалне кругоūраве*). Оно што је било изненађујуће је да се испоставило да је веома тешко формално доказати ово тврђење и било га је могуће доказати само када је формализовано дејство Мебијусових трансформација на кругоправе, што је омогућило да се користи „бгно" резоновање. Приметимо да не постоје имагинарне праве јер кад је $A = 0$, онда `mat_det` $H \geq 0$.

Коначно, веза између реалних кругоправих и еуклидских правих и кругова се може успоставити.

```
lemma
  assumes "is_circle H" "(μ, r) = euclidean_circle H"
  shows "circline_set H = of_complex ` {z. |z − μ| = r}"
lemma
  assumes "is_line H" "(z₁, z₂) = euclidean_line H"
          "circline_type H < 0"
  shows
  "circline_set H - {∞hc} = of_complex ` {z. collinear z₁ z₂ z}"
```

Приметимо да прва лема такође важи за имагинарни и тачка круг јер су оба скупа празна. Ипак, друга лема једино важи за реалне праве јер у случају тачка праве важи да $z_1 = z_2$, па је леви скуп празан, а десни је универзални скуп.

Кругоправе на Римановој сфери.

Кругоправе у равни одговарају круговима на Римановој сфери, и ми смо формално доказали ову везу. Сваки круг у тродимензионом простору се може добити као пресек сфере и равни. Успоставили смо један-на-један пресликавање између кругова на Римановој сфери и равни у простору. Приметимо и

да није неопходно да раван сече сферу и тада ћемо рећи да она дефинише јединствен имагинаран круг. Веза између равни у простору и кругоправих у проширеној комплексној равни је описао Швердфегер [148]. Ипак, аутор није приметио да за једну специјалну кругоправу (ону чија је репрезентативна матрица јединична матрица) не постоји раван у $\mathbb{R}^3$ која јој одговара — и да бисмо могли да имамо такву раван, потребно је да уместо посматрања равни у $\mathbb{R}^3$, узмемо у обзир тродимензионални пројективни простор и коначну хипераван. Зато, ми дефинишемо раван на следећи начин (опет у три корака).

```
typedef R4_vec≠0 = "{(a, b, c, d) :: R4_vec. (a, b, c, d) ≠ vec_zero}"
```

Приметимо да ће у $\mathbb{R}^3$, један од бројева a, b, или c бити различит од 0. Ипак, наша дефиниција дозвољава постојање равни $(0, 0, 0, d)$ која лежи у бесконачности. Функција репрезентације ће бити означена са $\lfloor\_\rfloor_{R4}$, а апстрактна функција ће бити означена са $\lceil\_\rceil^{R4}$. Поново, две равни су еквивалентне акко су пропорционалне (овог пута за неки ненула реални фактор).

```
definition ≈R4 :: "R4_vec≠0 ⇒ R4_vec≠0 ⇒ bool" where
  "α1 ≈R4 α2 ⟷ (∃k. k ≠ 0 ∧ ⌊α2⌋R4 = k * ⌊α1⌋R4)"
```

Коначно, равни (кругови који су у њима су добијени пресеком са Римановом сфером) се дефинишу као класа еквиваленције ове релације.

```
quotient_type plane = R4_vec≠0 / ≈R4
```

Коефицијенти равни дају линеарну једначину а тачка на Римановој сфери лежи на кругу одређеном са равни акко њена репрезентација задовољава линеарну једначину.

```
definition on_sphere_circle_rep where
  "on_sphere_circle_rep α M ⟷
    (let (a, b, c, d) = ⌊α⌋R4; (X, Y, Z) = ⌊M⌋R3
      in a * X + b * Y + c * Z + d = 0)"
lift_definition on_sphere_circle ::
  "plane ⇒ riemann_sphere ⇒ bool" is on_sphere_circle_rep
definition sphere_circle_set :: "riemann_sphere set" where
  "sphere_circle_set α = {A. on_sphere_circle α A}"
```

Приметимо да нисмо морали да уведемо тачке у тродимензионом пројективном простору (и њихове хомогене координате) јер смо ми једино заинтересовани за тачке на Римановој сфери које нису бесконачне.

Следеће, ми уводимо стереографску и инверзну стереографску пројекцију између кругова на Римановој сфери и кругова у проширеној комплексној равни.

```
definition stereographic_circline_rep where
  "stereographic_circline_rep α =
    (let (a,b,c,d) = ⌊α⌋_R4; A = cor((c+d)/2); B = (cor a + i * cor b)/2);
         C = (cor a − i * cor b)/2; D = cor((d−c)/2))
      in ⌈(A,B,C,D)⌉^H"
lift_definition stereographic_circline :: "plane ⇒ circline" is
  stereographic_circline_rep
definition inv_stereographic_circline_rep where
  "inv_stereographic_circline_rep H =
    (let (A,B,C,D) = ⌊H⌋_H
      in ⌈(Re(B+C), Re(i*(C−B)), Re(A−D), Re(D+A))⌉^R4"
lift_definition inv_stereographic_circline :: "circline ⇒ plane" is
  inv_stereographic_circline_rep
```

Ова два пресликавања су бијективна и међусобно инверзна. Пројекција скупа тачака круга на Римановој сфери је управо скуп тачака на кругоправој која се добија управо уведеном стереографском пројекцијом круга.

```
lemma "stereographic_circline ∘ inv_stereographic_circline = id"
lemma "inv_stereographic_circline ∘ stereographic_circline = id"
lemma "bij stereographic_circline" "bij inv_stereographic_circline"
lemma "stereographic ` sphere_circle_set α =
       circline_set (stereographic_circline α)"
```

Риманове кругоправе.

Још једна интересантна чињеница је да су реалне кругоправе ништа друго до скупови тачака које су на једнаком одстојању од неких датих тачака (заправо увек постоје тачно две такве тачке), али посматрајући одстојање у тетивној метрици. На Римановој сфери ове две тачке (звaћемо их тетивни

центри) се добијају пресеком сфере и праве која пролази кроз центар круга и нормална је на раван која садржи тај круг.

Тетивна кругоправа са датом тачком a и полупречником r је одређена на следећи начин.

```
definition chordal_circle_rep where "chordal_circle_rep μc rc =
  (let (μ1, μ2) = ⌊μc⌋C2;
       A = 4*|μ2|² - (cor rc)²*(|μ1|² + |μ2|²); B = −4*μ1*cnj μ2;
       C = -4*cnj μ1*μ2; D = 4*|μ1|² - (cor rc)²*(|μ1|² + |μ2|²)
    in mk_circline_rep A B C D)"
lift_definition chordal_circle :: "complex_hc ⇒ real ⇒ circline" is
  chordal_circle_rep
lemma "z ∈ circline_set (chordal_circle μc rc) ⟷
       rc ≥ 0 ∧ dist_hc z μc = rc"
```

За дату кругоправу, њен центар и радијус се могу одредити ослањајући се на следеће леме (у зависности да ли су коефицијенти B и C у репрезентативној матрици једнаки нули).

```
lemma
 assumes "is_C2_mat_herm (A, B, C, D)" "Re (A * D) < 0" "B = 0"
 shows
   "mk_circline A B C D =
            chordal_circle ∞hc sqrt(Re ((4 * A)/(A − D)))"
   "mk_circline A B C D =
            chordal_circle 0hc sqrt(Re ((4 * D)/(D − A)))"
lemma
 assumes "Re (mat_det (A, B, C, D)) < 0" "B ≠ 0"
         "is_C2_mat_herm (A, B, C, D)" "C * μc² + (D − A) * μc − B = 0"
         "rc = sqrt((4 + Re((4 * μc/B) * A))/(1 + Re(|μc|²)))"
 shows "mk_circline A B C D = chordal_circle (of_complex μc) rc"
```

Као и у претходним случајевима, може се увести функција која враћа тетивне параметре (потребно је направити разлику међу случајевима $B = 0$ и $B \neq 0$ и у другом случају је потребно решити квадратну једначину која описује тетивни центар).

Симетрија. Још од античке Грчке, инверзија круга је посматрана као аналогија осној рефлексији. У проширеној комплексној равни не постоји суштинска разлика између кругова и правих, тако да ћемо ми посматрати само једну врсту релације и за две тачке ћемо рећи да су *симетричне у односу на круг* ако се оне сликају једна у другу коришћењем било рефлексије или инверзије у односу на произвољну праву или круг. Када смо тражили алгебраску репрезентацију ове релације изненадили смо се колико је била једноставна и елегантна – тачке су симетричне акко је билинеарна форма њиховог репрезентативног вектора и репрезентативне матрице кругоправе једнака нули.

```
definition circline_symmetric_rep where
  "circline_symmetric_rep z1 z2 H ⟷
              bilinear_form ⌊z1⌋C2 ⌊z2⌋C2 ⌊H⌋H = 0"
lift_definition circline_symmetric :: "complexhc ⇒ complexhc ⇒
  circline ⇒ bool" is circline_symmetric_rep
```

Посматрајући скуп тачака на кругоправој и поредећи наше две дефиниције, постаје јасно да тачке на кругоправој су управо оне које су инваријантне у односу на симетрију у односу на ту кругоправу.

```
lemma "on_circline H z ⟷ circline_symmetric H z z"
```

Дејство Мебијусових трансформација на кругоправе.

Већ смо видели како Мебијусове трансформације делују на тачке $\overline{\mathbb{C}}$. Оне такође делују и на кругоправе (и дефиниција је изабрана тако да су два дејства компатибилна). Додатно, дајемо и дефиницију сличности две матрице (која је дефинисана у нашој помоћној теорији линеарне алгебре).

```
definition "congruence M H = mat_adj M *mm H *mm M"
definition mobius_circline_rep ::
  "C2_mat_reg ⇒ C2_mat_herm ⇒ C2_mat_herm" where
  "mobius_circline_rep M H = ⌈congruence (mat_inv ⌊M⌋M) ⌊H⌋H⌉H"
lift_definition mobius_circline :: "mobius ⇒ circline ⇒ circline"
  is mobius_circline_rep
```

Својства која има дејство Мебијусових трансформација на кругоправе је врло слично као и код дејства Мебијусових трансформација на тачке. На пример,

```
lemma "mobius_circline (mobius_comp M₁ M₂) =
       mobius_circline M₁ ∘ mobius_circline M₂"
lemma "mobius_circline (mobius_inv M) = inv (mobius_circline M)"
lemma "mobius_circline (mobius_id) = id"
lemma "inj mobius_circline"
```

Централна лема у овом одељку успоставља везу између дејства Мебијусових трансформација на тачкама и на кругоправама (и што је основно, доказује се да Мебијусове трансформације сликају кругоправе на кругоправе).

```
lemma "mobius_pt M ` circline_set H =
       circline_set (mobius_circline M H)"
```

Поред овога чува се и тип кругоправе (што повлачи, на пример, да се реалне кругоправе сликају на реалне кругоправе).

```
lemma "circline_type (mobius_circline M H) = circline_type H"
```

Још једно важно својство (које је нешто општије него претходно наведено) је да је симетрија тачака очувана након дејства Мебијусових трансформација (што се још назива и *принцип симетрије*).

```
lemma assumes "circline_symmetric z₁ z₂ H"
      shows "circline_symmetric (mobius_pt M z₁) (mobius_pt M z₂)
                                (mobius_circline M H)"
```

Последње две леме су веома важни геометријски резултати, и захваљујући веома погодној алгебарској репрезентацији, њих је било прилично лако доказати у нашој формализацији. Оба доказа се заснивају на следећој једноставној чињеници из линеарне алгебре.

```
lemma "mat_det M ≠ 0 ⟹ bilinear_form z₁ z₂ H =
   bilinear_form (M *mv z₁) (M *mv z₂) (congruence (mat_inv M) H)"
```

Јединственост кругоправе.

У еуклидској геометрији добро је позната чињеница да постоји јединствена права кроз две различите тачке и јединствени круг кроз три неколинеарне различите тачке. Слични резултати важе и у $\overline{\mathbb{C}}$. Ипак, да би се дошло до закључака потребно је извршити анализу случајева према типу кругоправе. Кругоправе озитивног типа не садрже тачке па код њих не постоји јединственост. Кругоправе нула типа садрже једну тачку и за сваку тачку постоји јединствена кругоправа нула типа која је садржи. Постоји јединствена кругоправа кроз било које три различите тачке (и она мора бити негативног типа).

lemma "∃! H. circline_type H = 0 ∧ z ∈ circline_set H"
lemma "⟦$z_1 \neq z_2$; $z_1 \neq z_3$; $z_2 \neq z_3$⟧ ⟹
∃! H. z_1 ∈ circline_set H ∧ z_2 ∈ circline_set H ∧
z_3 ∈ circline_set H"

Веома изненађујуће, директно доказивање ових лема је било веома тешко. Ипак, након примене „бгно" резоновања и након пресликавања тачака у канонску позицију (0_{hc}, 1_{hc} и ∞_{hc}) добили смо веома кратак и елегантан доказ (јер је могуће доказати, коришћењем израчунавања, да је x-оса једина кругоправа кроз ове три канонске тачке). Како су праве карактеризоване као управо оне кругоправе које садрже ∞_{hc}, постаје јасно да постоји јединствена права кроз било које две различите коначне тачке.

Скуп кардиналности кругоправе. Још једна од ствари која се узима „здраво за готово" је кардиналност кругоправи различитог типа. Већ смо рекли да ови докази захтевају „бгно" резоновање, али овог пута користили смо другачију врсту „бгно" резоновања. Испоставља се да је у многим случајевима лакше резоновати о круговима уколико је њихов центар у координатном почетку — у том случају, њихова матрица је дијагонална. Ми смо формализовали специјалан случај чувеног резултата из линеарне алгебре да је 2×2 хермитска матрица слична са реалном дијагоналном матрицом (штавише, елементи на дијагонали су реалне сопствене вредности матрице, а подударност је успостављена коришћењем унитарних матрица — подударност се такође може успоставити коришћењем једноставније матрице (матрице транслације), али онда она не би имала многа лепа својства).

```
lemma assumes "hermitean H"
  shows "∃k1 k2 M. mat_det M ≠ 0 ∧ unitary M ∧
                   congruence M H = (cor k1, 0, 0, cor k2)"
```

Последица је да за сваку кругоправу постоји унитарна Мебијусова трансформација која слика кругоправу тако да је њен центар у координатном почетку (заправо, постоје две такве трансформације ако су сопствене вредности различите). Видећемо да унитарне трансформације одговарају ротацијама Риманове сфере, тако да последња чињеница има једноставно геометријско објашњење. Кругоправе се могу дијагонализовати коришћењем само транслација, али унитарне трансформације често имају лепша својства.

```
lemma "∃ M H'. unitary_mobius M ∧
                 mobius_circline M H = H' ∧ circline_diag H'"
lemma assumes "⋀ H'. circline_diag H' ⟹ P H"
              "⋀ M H. P H ⟹ P (mobius_circline M H)"
      shows   "P H"
```

Приметимо да је `unitary_mobius` предикат који подиже `unitary` својство са $\mathbb{C}^2$ матрица на тип `mobius`. Слично, `circline_diag` подиже услов дијагоналне матрице на тип `circline`.

Коришћењем овакве врсте „бгно" резоновања постаје прилично јасно како доказати следећу карактеризацију за кардиналност скупа кругоправе.

```
lemma "circline_type H > 0 ⟷ circline_set H = {}"
lemma "circline_type H = 0 ⟷ ∃z. circline_set H = {z}"
lemma "circline_type H < 0 ⟷
   ∃z1 z2 z3. z1 ≠ z2 ∧ z1 ≠ z3 ∧ z2 ≠ z3 ∧ circline_set H ⊇ {z1, z2, z3}"
```

Важна, нетривијална, последица јединствености кругоправе и кардиналности скупа кругоправе је да је функција `circline_set` инјективна, тј. за сваки непразан скуп тачака кругоправе, постоји јединствена класа пропорционалних матрица која их све одређује (`circline_set` је празан за све имагинарне кругоправе, што значи да ово својство не важи када је скуп тачака кругоправе празан).

```
lemma "⟦ circline_set H₁ = circline_set H₂; circline_set H₁ ≠ {} ⟧
  ⟹ H₁ = H₂"
```

Оријентисане кругоправе

У овом одељку ми ћемо описати како је могуће увести оријентацију за кругоправе. Многи важни појмови зависе од оријентације. Један од најважнијих појмова је појам *диска* — унутрашњост кругоправе. Слично као што је то био случај код скупа тачака кругоправе, скуп тачака диска се уводи коришћењем квадратне форме у чијем изразу се налази матрица кругоправе — скуп тачака диска кругоправе је скуп тачака за које важи $A*z*\texttt{cnj}\ z+B*\texttt{cnj}\ z+C*z+D < 0$, при чему је $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$ матрица која репрезентује кругоправу. Како скуп тачака диска мора бити инваријантан у односу на избор представника, јасно је да су матрице оријентисане кругоправе еквивалентне само ако су оне пропорционалне у односу на неки реални фактор (подсетимо се да код неоријентисаних кругоправих фактор може бити произвољан реалан ненула број).

```
definition ≈ocm :: "C2_mat_herm ⇒ C2_mat_herm ⇒ bool" where
  "H₁ ≈ocm H₂ ⟷ (∃ (k::real). k > 0 ∧ ⌊H₂⌋H = cor k *sm ⌊H₁⌋H)"
```

Лако се доказује да је ова дефинисана релација релација еквиваленције, тако да су кругоправе дефинисане преко количничке конструкције као класе еквиваленције.

```
quotient_type o_circline = C2_mat_herm / ≈ocm
```

Сада можемо користити квадратну форму да дефинишемо унутрашњост, спољашњост и границу оријентисане кругоправе.

```
definition on_o_circline_rep :: "C2_mat_herm ⇒ C2_vec≠0 ⇒ bool"
  where "on_o_circline_rep H z ⟷ quad_form ⌊H⌋H ⌊z⌋C2 = 0"
definition in_o_circline_rep :: "C2_mat_herm ⇒ C2_vec≠0 ⇒ bool"
  where "in_o_circline_rep H z ⟷ quad_form ⌊H⌋H ⌊z⌋C2 < 0"
definition out_o_circline_rep :: "C2_mat_herm ⇒ C2_vec≠0 ⇒ bool"
  where "out_o_circline_rep H z ⟷ quad_form ⌊H⌋H ⌊z⌋C2 > 0"
```

Ове дефиниције се подижу на `on_o_circline`, `in_o_circline`, и `out_o_circline` (при томе доказујемо неопходне услове), и, коначно, уводе се следеће три дефиниције.

```
definition o_circline_set :: "complex_hc set" where
  "o_circline_set H = {z. on_o_circline H z}"
definition disc :: "complex_hc set" where
  "disc H = {z. in_o_circline H z}"
definition disc_compl :: "complex_hc set" where
  "disc_compl H = {z. out_o_circline H z}"
```

Ова три скупа су међусобно дисјунктна и заједно испуњавају целу раван.

```
lemma "disc H ∩ disc_compl H = {}"
      "disc H ∩ o_circline_set H = {}"
      "disc_compl H ∩ o_circline_set H = {}"
      "disc H ∪ disc_compl H ∪ o_circline_set H = UNIV"
```

За дату оријентисану кругоправу, може се тривијално одредити њен неоријентисани део, а ове две кругоправе имају исти скуп тачака.

```
lift_definition of_o_circline (_∘) :: "o_circline ⇒ circline" is id
lemma "circline_set (H∘) = o_circline_set H"
```

У **lift_definition** увели смо краћи запис функције `of_o_circline`, тако да, на пример, H° у леми је скраћеница за `of_o_circline` H.

За сваку кругоправу, постоји тачно једна супротно оријентисана кругоправа.

```
definition "opp_o_circline_rep H = ⌈-1 *sm ⌊H⌋_H⌉^H"
lift_definition opp_o_circline (_↔) :: "o_circline ⇒ o_circline" is
  opp_o_circline_rep
```

Одређивање супротне кругоправе је идемпотентно јер супротне кругоправе имају исти скуп тачака, али размењују диск и његов комплемент.

```
lemma "(H↔)↔ = H"
lemma "o_circline_set (H↔) = o_circline_set H"
      "disc (H↔) = disc_compl H" "disc_compl (H↔) = disc H"
```

Функције _$^{\circ}$ и o_circline_set су у одређеном смислу инјективне.

```
lemma "H₁° = H₂° ⟹ H₁ = H₂ ∨ H₁ = H₂↔"
lemma
  "⟦o_circline_set H₁ = o_circline_set H₂; o_circline_set H₁ ≠ {}⟧
   ⟹ H₁ = H₂ ∨ H₁ = H₂↔"
```

Дата хермитска матрица кругоправе представља тачно једну од две могуће оријентисане кругоправе. Избор шта ћемо звати позитивно оријентисана кругоправа је произвољан. Ми смо одлучили да пратимо приступ који је предложио Швердфегер [148], где се користи водећи коефицијент A као први критеријум, који каже да се кругоправе са матрицом у којој важи $A > 0$ зову позитивно оријентисане, а ако у матрици важи $A < 0$ онда се зову негативно оријентисане. Ипак, Швердфегер није дискутовао још један могући случај када је $A = 0$ (случај правих), тако да смо ми морали да проширимо његову дефиницију да би имали потпуну карактеризацију.

```
definition pos_o_circline_rep where "pos_o_circline_rep H ⟷
  (let (A, B, C, D) = ⌊H⌋_H
    in Re A > 0 ∨
       (Re A = 0 ∧ ((B ≠ 0 ∧ arg B > 0) ∨ (B = 0 ∧ Re D > 0))))"
lift_definition pos_o_circline :: "o_circline ⇒ bool"
  is pos_o_circline_rep
```

Сада, тачно једна од две супротно оријентисане кругоправе је позитивно оријентисана.

```
lemma "pos_o_circline H ∨ pos_o_circline (H↔)"
      "pos_o_circline (H↔) ⟷ ¬ pos_o_circline H"
```

Оријентација кругова је и алгебарски једноставна (посматра се знак коефицијента A) и геометријски природна захваљујући следећој једноставној карактеризацији.

```
lemma "∞ₕ ∉ o_circline_set H ⟹
       pos_o_circline H ⟷ ∞ₕ ∉ disc H"
```

Још једна лепа геометријска карактеризација за позитивну оријентацију је да је еуклидски центар позитивно оријентисаних еуклидових кругова садржан у њиховом диску.

```
lemma assumes "is_circle (H∘)" "circline_type (H∘) < 0"
              "(a, r) = euclidean_circle (H∘)"
      shows "pos_oriented H ⟷ of_complex a ∈ disc H"
```

Приметимо да је оријентација правих и тачака кругова вештачки уведена (само да бисмо имали тотално дефинисану позитивну оријентацију), и она нема природну геометријску интерпретацију. Због овога, непрекидност оријентације је прекинута и ми мислимо да није могуће увести оријентацију правих тако да функција оријентације буде свуда непрекидна. Зато, када у неким наредним лемама будемо говорили о оријентацији ми ћемо експлицитно искључити случај правих.

Тотална карактеризација за позитивну оријентацију нам омогућава да створимо пресликавање из неоријентисаних у оријентисане кругоправе (добијамо увек позитивно оријентисане кругоправе).

```
definition of_circline_rep :: "C2_mat_herm ⇒ C2_mat_herm" where
  "of_circline_rep H = (if pos_o_circline_rep H then H
                        else opp_o_circline_rep H)"
lift_definition of_circline (_∘) :: "circline ⇒ o_circline" is
  of_circline_rep
```

Доказана су бројна својства функције `of_circline`, а овде ћемо навести само најзначајнија.

```
lemma "o_circline_set (H∘) = circline_set H"
lemma "pos_o_circline (H∘)"
lemma "(H∘)∘ = H" "pos_o_circline H ⟹ (H∘)∘ = H"
lemma "H1∘ = H2∘ ⟹ H1 = H2"
```

Дејство Мебијусових трансформација на оријентисане кругоправе. На репрезентативном нивоу дејство Мебијусових трансформација на оријентисане кругоправе је исто као и дејство на неоријентисане кругоправе.

```
lift_definition mobius_o_circline ::
  "mobius ⇒ o_circline ⇒ o_circline" is mobius_circline_rep
```

Дејство Мебијуса на (неоријентисане) кругоправе се може дефинисати коришћењем дефиниције за дејство Мебијуса на оријентисане кругоправе, али обрнуто не би могло.

```
lemma "mobius_circline M H = (mobius_o_circline M (H^↻))^○"
lemma "let H₁ = mobius_o_circline M H;
           H₂ = (mobius_circline M (H^○))^↻
       in H₁ = H₂ ∨ H₁ = H₂^↔"
```

Дејство Мебијусових трансформација на оријентисане кругоправе има слична својства као и дејство Мебијусових трансформација на неоријентисане кругоправе. На пример, оне се слажу у погледу инверза (`lemma "mobius_o_circline (mobius_inv M) = inv (mobius_o_circline M)"`), композиције, идентитета, обе су инјективне (`inj mobius_circline`), и тако даље. Централне леме у овом одељку повезују дејства Мебијусових трансформација на тачкама, оријентисаним кругоправама и дисковима.

```
lemma "mobius_pt M ` o_circline_set H =
       o_circline_set (mobius_o_circline M H)"
lemma "mobius_pt M ` disc H = disc (mobius_o_circline M H)"
lemma "mobius_pt M ` disc_compl H =
       disc_compl (mobius_o_circline M H)"
```

Све еуклидске сличности чувају оријентацију кругоправе.

```
lemma assumes "a ≠ 0" "M = similarity a b"
              "∞_hc ∉ o_circline_set H"
  shows
    "pos_o_circline H ⟷ pos_o_circline (mobius_o_circline M H)"
```

Оријентација слике дате оријентисане кругоправе H након дате Мебијусове трансформације M зависи од тога да ли пол M (тачка коју трансформација M слика у ∞_{hc}) лежи на диску или у диску који је комплементаран H (ако је у скупу H, онда се слика у праву, а у том случају не дискутујемо оријентацију).

```
lemma
 "0_hc ∈ disc_compl H ⟹
   pos_o_circline (mobius_o_circline reciprocation H)"
 "0_hc ∈ disc H ⟹
   ¬ pos_o_circline (mobius_o_circline reciprocation H)"
lemma
 assumes "M = mk_mobius a b c d" "c ≠ 0" "a*d - b*c ≠ 0"
 shows "pole M ∈ disc H ⟶
        ¬ pos_o_circline (mobius_o_circline M H)"
       "pole M ∈ disc_compl H ⟶
        pos_o_circline (mobius_o_circline M H)"
```

Приметимо да је ово другачије него што тврди Швердфегер [148]: „Реципроцитет чува оријентацију круга који не садржи 0, али инвертује оријентацију било ког круга који садржи 0 као унутрашњу тачку. Свака Мебијусова трансформација чува оријентацију било ког круга који не садржи свој пол. Ако круг садржи свој пол, онда круг који се слика има супротну оријентацију". Наша формализација доказује да оријентација резултујућег круга не зависи од оријентације полазног круга (на пример, у случају реципроцитета, оријентација полазног круга показује релативну позицију круга и тачке бесконачно што је одређено знаком коефицијента A у репрезентативној матрици и то је сасвим независно од релативне позиције круга и нула тачке које су одређене знаком коефицијента D — ова два коефицијента се размењују приликом примене трансформације реципроцитета).

Очување угла

Мебијусове трансформације су конформно пресликавање, што значи да оне чувају оријентисане углове међу оријентисаним кругоправама. Ако се угао дефинише коришћењем чисто алгебарског приступа (пратећи [148]), онда је врло лако доказати ово својство. Поред дефиниције угла, навешћемо и дефиницију мешовите детерминанте коју смо дефинисали раније у нашој основној теорији.

```
fun mat_det_mix :: "C2_mat ⇒ C2_mat ⇒ complex" where
 "mat_det_mix (A1, B1, C1, D1) (A2, B2, C2, D2) =
    A1 * D2 - B1 * C2 + A2 * D1 - B2 * C1"
```

```
definition cos_angle_rep where
 "cos_angle_rep H1 H2 =
        - Re (mat_det_mix ⌊H1⌋H ⌊H2⌋H) /
        2 * (sqrt (Re (mat_det ⌊H1⌋H * mat_det ⌊H2⌋H))))"
lift_definition cos_angle :: "o_circline ⇒ o_circline ⇒ complex"
  is cos_angle_rep
lemma "cos_angle H1 H2 =
  cos_angle (moebius_o_circline M H1) (moebius_o_circline M H_2)"
```

Ипак, ова дефиниција није интуитивна, и из педагошких разлога желели смо да је повежемо са нешто уобичајенијом дефиницијом. Прво, дефинисали смо угао између два комплексна вектора ($\lfloor \_ \rfloor$ означава функцију за нормализацију угла која је описана раније).

```
definition ang_vec ("∡") where "∡ z1 z2 = ⌊arg z2 - arg z1⌋"
```

За дати центар μ обичног еуклидског круга и тачку z на њему, дефинишемо тангентни вектор у z као радијус вектор $\overrightarrow{\mu z}$, ротиран за $\pi/2$, у смеру казаљке на сату или у супротном смеру у зависности од оријентације.

```
definition tang_vec :: "complex ⇒ complex ⇒ bool ⇒ complex" where
  "tang_vec μ z p = sgn_bool p * i * (z - μ)"
```

У логичкој променљивој p енкодира се оријентација круга, а функција `sgn_bool` p враћа 1 када је p тачно, а -1 када је p нетачно. Коначно, угао између два оријентисана круга у њиховој заједничкој тачки z се дефинише као угао између тангентних вектора у z.

```
definition ang_circ where
  "ang_circ z μ1 μ2 p1 p2 = ∡ (tang_vec μ1 z p1) (tang_vec μ2 z p2)"
```

Коначно, веза између алгебарске и геометријске дефиниције косинуса угла дата је следећом лемом.

```
lemma assumes "is_circle (H1°)" "is_circle (H2°)"
    "circline_type (H1°) < 0" "circline_type (H2°) < 0"
    "(μ1, r1) = euclidean_circle (H1°)"
    "(μ2, r2) = euclidean_circle (H2°)"
    "of_complex z ∈ o_circline_set H1 ∩ o_circline_set H2"
```

```
shows "cos_angle H1 H2 =
  cos (ang_circ z μ1 μ2 (pos_o_circline H1) (pos_o_circline H2))"
```

Да бисмо доказали ову лему било је неопходно доказати закон косинуса у систему *Isabelle/HOL*, али се ово показало као веома једноставан задатак.

5.7 Неке важне подгрупе Мебијусових трансформација

Већ смо описали параболичку групу (групу еуклидских сличности), кључну за еуклидску геометрију равни. Сада ћемо описати карактеристике две веома важне подгрупе Мебијусове групе — групу сферних ротација, важну за елиптичку планарну геометрију, и групу аутоморфизама диска која је важна за хиперболичку планарну геометрију.

Ротације сфере. Генерална унитарна група, коју означавамо са $GU_2(\mathbb{C})$ је група која садржи све Мебијусове трансформације које су репрезентоване уопштеним унитарним матрицама.

```
definition unitary_gen where
 "unitary_gen M ⟷
  (∃ k::complex.  k ≠ 0 ∧ mat_adj M *mm M = k *sm eye)"
```

Иако је у дефиницији дозвољено да k буде комплексан фактор, испоставља се да је једино могуће да k буде реалан. Генерализоване унитарне матрице могу бити растављене на обичне унитарне матрице и јединичне матрице које су помножене неким позитивним фактором.

```
definition unitary where "unitary M ⟷ mat_adj M *mm M = eye"
lemma "unitary_gen M ⟷
  (∃ k M'.  k > 0 ∧ unitary M' ∧ M = (cor k *sm eye) *mm M')"
```

Група унитарних матрица је веома важна јер описује све ротације Риманове сфере (изоморфна је реалној специјалној ортогоналној групи $SO_3(\mathbb{R})$). Једна од могућих карактеризација $GU_2(\mathbb{C})$ у $\overline{\mathbb{C}}$ је да је то група трансформација таквих да је имагинарни јединични круг фиксан (ово је круг чија је матрица репрезентације јединична и налази се у равни у бесконачности).

```
lemma "mat_det (A,B,C,D) ≠ 0 ⟹ unitary_gen (A, B, C, D) ⟷
  moebius_circline (mk_moebius A B C D) imag_unit_circle =
  imag_unit_circle"
```

Карактеризација генерализованих унитарних матрица у координатама је дата следећом лемом.

```
lemma "unitary_gen M ⟷ (∃ a b k. let M' = (a, b, −cnj b, cnj a) in
  k ≠ 0 ∧ mat_det M' ≠ 0 ∧ M = k *sm M')"
```

Додатно, дефинисали смо специјалну унитарну групу $SU_2(\mathbb{C})$, која садржи генерализоване унитарне матрице са детерминантом једнаком један (оне се препознају по форми $(a, b, -\texttt{cnj}\ b, \texttt{cnj}\ a)$), без множитеља k, и ову специјалну групу користимо да бисмо извели координатну форму генерализованих унитарних матрица.

Аутоморфизми диска. Дуална група претходној групи трансформација је група генерализованих унитарних матрица чија сигнатура је $1-1$ ($GU_{1,1}(\mathbb{C})$).

```
definition unitary11 where
  "unitary11 M ⟷ mat_adj M *mm (1,0,0,−1) *mm M = (1,0,0,−1)"
definition unitary11_gen where
  "unitary11_gen M ⟷ (∃ k::complex. k ≠ 0 ∧
    mat_adj M *mm(1,0,0,−1)*mm M = k *sm (1,0,0,−1))"
```

Поново, дефиниција дозвољава комплексан фактор k, али се показује да једино реални фактори имају смисла.

Карактеризација $GU_{1,1}(\mathbb{C})$ је да она садржи све Мебијусове трансформације које фиксирају јединични круг.

```
lemma "mat_det (A,B,C,D) ≠ 0 ⟹ unitary11_gen (A, B, C, D) ⟷
  moebius_circline (mk_moebius A B C D) unit_circle = unit_circle"
```

Карактеризација генерализоване унитарне 1-1 матрице у координатама је дата следећим лемама.

```
lemma "unitary11_gen M ⟷ (∃ a b k. let M' = (a, b, cnj b, cnj a) in
  k ≠ 0 ∧ mat_det M' ≠ 0 ∧
```

$(M = k *_{sm} M' \ \vee \ M = k *_{sm} (\texttt{cis}\ \ pi,\ 0,\ 0,\ 1) *_{sm} M')$"

lemma "unitary11_gen M $\longleftrightarrow$ $(\exists\ a\ b\ k.$ **let** $M' = (a,\ b,\ \texttt{cnj}\ \ b,\ \texttt{cnj}\ \ a)$ **in**
$k \neq 0\ \wedge\ \texttt{mat\_det}\ \ M' \neq 0\ \wedge\ M = k *_{sm} M'\ )$"

Приметимо да је прва лема садржана у другој леми. Ипак, било је лакше доказати прву лему јер добијамо матрице следећег облика $k *_{sm} (a,\ b,\ -cnj\ b,\ -cnj\ a)$ — геометријски, друга група трансформација комбинује прву групу са додатном централном симетријом.

Још једна важна карактеризација ових трансформација је коришћењем такозваног Блашке фактора. Свака трансформација је композиција Блашке фактора (рефлексије која неку тачку која је на јединичној кружници слика у нула) и ротације.

lemma assumes "$k \neq 0$" "$M' = (a,\ b,\ \texttt{cnj}\ \ b,\ \texttt{cnj}\ \ a)$"
"$M = k *_{sm} M'$" "$\texttt{mat\_det}\ \ M' \neq 0$" "$a \neq 0$"
shows "$\exists\ k'\ \phi\ a'.\ \ k' \neq 0\ \wedge\ a' * \texttt{cnj}\ \ a' \neq 1\ \wedge$
$M = k' *_{sm} (\texttt{cis}\ \ \phi,\ 0,\ 0,\ 1) *_{mm} (1,\ -a',\ -\texttt{cnj}\ \ a',\ 1)$"

Изузетак је у случају када је $a = 0$ и онда се уместо Блашке фактора, користи реципроцитет (бесконачно замењује a' у претходној леми).

lemma assumes "$k \neq 0$" "$M' = (0,\ b,\ \texttt{cnj}\ \ b,\ 0)$" "$b \neq 0$" "$M = k *_{sm} M'$"
shows "$\exists\ k'\ \phi.\ \ k' \neq 0\ \wedge\ M = k' *_{sm} (\texttt{cis}\ \ \phi,\ 0,\ 0,\ 1) *_{mm} (0,\ 1,\ 1,\ 0)$"

Матрице $GU_{1,1}(\mathbb{C})$ се природно деле у две подгрупе. Све трансформације фиксирају јединични круг, али прва подгрупа се састоји од трансформација које мапирају јединични диск у самог себе (такозвани *аутоморфизми диска*), док се друга подгрупа састоји из трансформација које размењују јединични диск и његов комплемент. За дату матрицу, њена подгрупа се једино може одредити посматрајући знак детерминанте $M' = (a,\ b,\ \texttt{cnj}\ \ b,\ \texttt{cnj}\ \ a)$. Ако је само $M = (a_1, b_1, c_1, d_1)$ дато, а нису дати M', а ни k, онда је критеријум за утврђивање подгрупе вредност $\texttt{sgn}(\texttt{Re}\ ((a_1 * d_1)/(b_1 * c_1)) - 1)$.

Приметимо да су све важне подгрупе овде описане једино у терминима алгебре. Формализовали смо и неке геометријске доказе који дају еквивалентну карактеризацију тврђењима које смо већ описали. Додатно, важи да су сви аналитички аутоморфизми диска једнаки композицији Блашке фактора и ротација (ипак, доказ се заснива на математичкој анализи, принципу

максималног модула и Шварцовој леми – техникама које ми нисмо узимали у обзир). Чак и слабије тврђење да су сви Мебијусови аутоморфизми диска ове форме није још формално доказано (кључни корак је доказати да аутоморфизми диска фиксирају јединични круг, а то је нешто што нисмо могли доказати без детаљног испитивања топологије на чему тренутно радимо).

Сличне Мебијусове трансформације и класификација Мебијусових трансформација

Да бисмо могли да класификујемо Мебијусове трансформације прво је било потребно увести пар нових појмова и анализирати њихова својства. Пре свега, анализирали смо фиксне тачке Мебијусових трансформација. Раније смо спомињали да су еуклидске сличности једине Мебијусове трансформације којима је ∞_{hc} фиксна тачка. Ипак, за доказе потребне у овом одељку морали смо да нешто више анализирамо фиксне тачке. Увели смо дефиницију фиксне тачке и дефиницију фиксне тачке која је коначна.

```
definition moebius_fixed_points where
  "moebius_fixed_points M γ ⟷ moebius_pt M γ = γ"
definition moebius_fixed_points_finite_rep where
  "moebius_fixed_points_finite_rep M γ ⟷
   (let (a, b, c, d) = ⌊M⌋_M
     in c * γ * γ - (a - d) * γ - b = 0)"
lift_definition moebius_fixed_points_finite ::
  "moebius ⇒ complex ⇒ bool" is
      moebius_fixed_points_finite_rep
```

За Мебијусову трансформацију могу постојати највише две фиксне тачке које могу бити и једнаке. Оне су обе коначне ако за коефицијент репрезентативне матрице важи $c \neq 0$; једна од њих је коначна, а једна бесконачна ако за коефицијенте важи $c = 0$ и $a \neq d$ и оне су обе једнаке бесконачно ако за коефицијенте важи $c = 0$ и $a = d$. Ово тврђење смо доказали у наредним лемама.

```
lemma
  assumes "mat_det (a, b, c, d) ≠ 0" "c ≠ 0"
  shows "∃γ1 γ2. moebius_fixed_points (mk_moebius a b c d) γ1 ∧
```

```
                moebius_fixed_points (mk_moebius a b c d) γ2 ∧
                γ1 ≠ ∞hc     ∧     γ2 ≠ ∞hc"
lemma
  assumes "mat_det (a, b, c, d) ≠ 0" "c = 0" "a ≠ d"
  shows "moebius_fixed_points (mk_moebius a b c d) ∞hc"
        "∃γ. moebius_fixed_points (mk_moebius a b c d) γ ∧ γ ≠ ∞hc"
lemma
  assumes "mat_det (a, b, c, d) ≠ 0" "c = 0" "a = d"
  shows "moebius_fixed_points (mk_moebius a b c d) ∞hc"
```

Ова тврђења није било тешко доказати, али су имала бројне кораке и случајеве које је требало размотрити и захтевала су доказивање доста ситних алгебарских корака.

Потом дефинишемо како се за две дате Мебијусове трансформације може одредити слична Мебијусова трансформација. Овде уједно дајемо и дефиницију сличних матрица коју смо увели и чија својства смо доказали у нашој основној теорији линеарне алгебре.

```
definition similarity_matrices where
  "similarity_matrices I M = I *mm M *mm mat_inv I"
definition moebius_mb_rep where
  "moebius_mb_rep I M = ⌈ similarity_matrices ⌊I⌋M ⌊M⌋M ⌉M"
lift_definition moebius_mb :: "moebius ⇒ moebius ⇒ moebius" is
  moebius_mb_rep
```

Сада је могуће дефинисати релацију сличности између две Мебијусове трансформације. Додатно, доказали смо и да је ово и релација еквиваленције, тј. доказано је да за релацију важи својство рефлексивности, симетричности и транзитивности и докази су били прилично директни и кратки.

```
definition similar where
  "similar M1 M2 ⟷ (∃I. moebius_mb I M1 = M2)"
lemma "similar M M"
lemma assumes "similar M1 M2"
  shows "similar M2 M1"
lemma assumes "similar M1 M2" "similar M2 M3"
  shows "similar M1 M3"
```

Врло важно тврђење је да је свака Мебијусова трансформација слична некој интегралној трансформацији (еуклидској сличности). Доказивање овог тврђења се свело на одређивање параметара еуклидксе сличности, a и b, за произвољну Мебијусову трансформацију. Да би одредили ове параметре било је потребно одредити фиксне тачке Мебијусове трансформације и користити својства за фиксне тачке које смо нешто раније доказали.

```
lemma
  "∃ k t. k ≠ 0 ∧ similar M (similarity a b)"
```

Веома важан параметар за Мебијусове трансформације јесте *инваријанта Мебијусових трансформација*. Она се дефинише коришћењем репрезентативне матрице за дату Мебијусову трансформацију, као однос између трага и детерминанте матрице. Потом, дефиницију са репрезентативног нивоа подижемо на ниво количничког типа.

```
definition similarity_invar_rep where
  "similarity_invar_rep M =
   (let M = ⌊M⌋_M in (mat_trace M)^2 / mat_det M - 4)"
lift_definition similarity_invar :: "moebius ⇒ complex" is
  similarity_invar_rep
```

Број 4 који се добија у дефиницији произилази из чињенице да cyS матрице идентичне трансформације сличне, тј. $E = I^{-1}EI$, где је $E = \lfloor$mobius_id$\rfloor_M$, а $I = \lfloor I \rfloor_M$. Како је mat_trace $E = 2$, а mat_det $E = 1$, њихов количник је 4.

Важно својство овог параметра је да су Мебијусове трансформације (које нису идентитет) сличне акко имају једнаке инваријанте.

```
lemma assumes "M₁ ≠ id_moebius" "M₂ ≠ id_moebius"
  shows
  "similarity_invar M₁ = similarity_invar M₂ ⟷ similar M₁ M₂"
```

Доказивање у једном смеру („ако су сличне имају једнаке инваријанте") било је једноставно и кратко. Међутим, супротан смер („ако имају једнаке инваријанте, онда су сличне") је представљао изазов, било је потребно раздвојити случајеве када је инваријанта једнака 0 и када је различита од 0, а потом у доказу су коришћене алгебарске трансформације, као и бројна својства релације

сличности матрица. Оно што је интересантно је да је доказ у Швердфегеру значајно краћи (само пар редова), мада се мора рећи да је аутор нотирао све важне тачке доказа, али је у формалном доказу било неопходно ући у дубљу анализу и сваку од ових тачака детаљније испитати.

Коначно, стижемо до класификације Мебијусових трансформација која се управо карактерише коришћењем инваријанте. За Мебијусову трансформацију кажемо да је то пресликавање које је:

параболичко,	$\texttt{similarity\_invar} = 0$, има само једну фиксну тачку
елиптичко,	инваријанта је реална и $-4 \leq \texttt{similarity\_invar} < 0$
правилно хиперболичко,	инваријанта је реална и $\texttt{similarity\_invar} > 0$
неправилно хиперболичко,	инваријанта је реална и $\texttt{similarity\_invar} \leq -4$
локсодромичко,	инваријанта није реална

5.8 Дискусија

Визуелно, геометријски аргументи се често користе у доказима у уџбеницима. Као пример, ми ћемо демонстрирати доказ о очувању својства угла након примене Мебијусових трансформација на који се често може наићи у различитим књигама о овој теми (у овом поглављу ми ћемо пратити Нидамов приступ [130] који има за циљ да представи област без формалних детаља, па самим тим књига није стриктно формално писана али, ипак, овакав начин резоновања присутан је и код многих других аутора).

Прво важно питање је појам угла. Углови могу бити дефинисани између оријентисаних или неоријентисаних кривих, а и сами углови могу бити оријентисани или неоријентисани. Нидам дефинише угао између две криве на следећи начин: „Нека су S_1 и S_2 криве које се секу у тачки z. Као што је илустровано, ми можемо повући њихове тангенте T_1 и T_2 у тачки z. Угао између кривих S_1 и S_2 у њиховој заједничкој тачки z је оштар угао α од T_1 до T_2. Значи овај угао α има знак који му је додељен: угао између S_2 и S_1 је минус илустровани угао између S_1 и S_2." То значи да је угао дефинисан само између неоријентисаних кривих (и то је различито у односу на нашу дефиницију),

али сам угао је оријентисан (а то је исто као и у нашој финалној дефиницији). У ранној фази наше формализације ми смо дефинисали и користили неоријентисани конвексан и оштар угао између два вектора.

```
definition "∡c" where "∡c z1 z2 ≡ abs (∡ z1 z2)"
definition acutize where "acutize α = (if α > π/2 then π - α else α)"
definition "∡a" where "∡a z1 z2 ≡ acutize (∡c z1 z2)"
```

Како су наше кругоправе оријентисане од старта, ми смо доказали да на оштар угао између два круга не утиче оријентација и да се он може изразити у терминима три тачке (тачке пресека и тачака које представљају центре кругова).

```
lemma "⟦z ≠ μ1 ; z ≠ μ2⟧ ⟹
       ang_circ_a z μ1 μ2 p1 p2 = ∡a (z − μ1) (z − μ2)"
```

Функција `ang_circ_a` је дефинисана као оштар угао између два тангентна вектора (слично функцији `ang_circ` у нашој коначној формализацији).

Доказ да Мебијусова трансформација чува угао који стоји у уџбенику [130] се ослања на чињеницу да се свака Мебијусова трансформација може растaвити на транслацију, ротацију, хомотетију и инверзију. Чињеница да транслације, ротације и дилетације чувају угао је узета као подразумевана и није доказивана (и да будемо искрени формализација ове чињенице није била тешка када смо успели да све појмове формално дефинишемо на одговарајући начин). Централни изазов је доказати да инверзија чува углове, тј. доказати тврђење „Инверзија је антикомфорно пресликавање". Доказ се заснива на „чињеници да за било коју дату тачку z која није на кругу инверзије K, постоји тачно један круг који је ортогоналан на K и пролази кроз z у било ком правцу". Даље, доказ се наставља са „Претпоставимо да се две криве S_1 и S_2 секу у z, и да су њихове тангенте T_1 и T_2, а угао између њих је α. Да бисмо сазнали шта се дешава са углом након инверзије у односу на K, заменимо S_1 и S_2 јединственим круговима R_1 и R_2 ортогоналним на K који пролазе кроз z у истом смеру као што је и смер S_1 и S_2, тј., круговима чије тангенте у z су T_1 и T_2. Како инверзија у односну на K слика сваки од ових кругова на саме себе нови угао у $\tilde{z}$ је $-\alpha$. Крај."

У нашем ранијем покушају ми смо формализовали овај „доказ", али је ово захтевало веома велику количину уложеног труда у поређењу са углађеним

алгебарским доказом у нашој финалној формализацији. Прво, уџбеник је често врло непрецизан у томе да ли се користи „комплексна инверзија" или „геометријска инверзија" (тј. према нашим терминима које смо раније увели – да ли се користи реципроцитет или инверзија). У доказу из уџбеника аутор користи инверзију у односу на произвољан круг K, али је довољно посматрати само реципроцитет (који је увек дат у односу на јединични круг). Формализација резоновања које је дато у уџбенику је већ дала прилично велике формуле, и било би још компликованије и монотоније (ако је уопште и могуће) завршити доказ коришћењем инверзије у односу на произвољни круг. На пример, једноставан реципроцитет круга са центром μ и радијусом r даје круг са центром $\tilde{\mu} = \mu/\texttt{cor}\ (|\mu|^2 - r^2)$, и радијусом $\tilde{r} = r/||\mu|^2 - r^2|$, и ова веза би била још комплекснија за произвољну Мебијусову трансформацију, ако би била записана у координатама, без коришћења појма матрица као што смо ми радили у нашој главној формализацији.

Формални запис тврђења о очувању угла је следећи.

```
lemma
  assumes "z ∈ circle μ1 r1" "z ∈ circle μ2 r2"
          "inv ` circle μ1 r1 = circle μ̃1 r̃1"
          "inv ` circle μ2 r2 = circle μ̃2 r̃2"
  shows "ang_circ_a z μ1 μ2 = ang_circ_a z̃ μ̃1 μ̃2"
```

Поред тога што недостаје дискусија за бројне специјалне случајеве, у неформалном доказу недостаје и један значајан део. Наиме, лако је доказати да је $\tilde{z}$ пресек R_1 и R_2 (то је пресек $\tilde{S}_1$ и $\tilde{S}_2$, које су слике S_1 и S_2 након инверзије), али доказати да R_1 и $\tilde{S}_1$ и да R_2 и $\tilde{S}_2$ имају исту тангенту у $\tilde{z}$ је захтевало не тако тривијална израчунавања (тај доказ се заснива на чињеници да су центар μ_i' круга R_i, центар $\tilde{\mu}_i$ круга $\tilde{S}_i$, и $\tilde{z}$ колинеарни).

Једноставан аргумент симетрије који каже да су углови између два круга у њиховим двема различитим тачкама пресека једнаки поново није било једноставно формализовати.

```
lemma assumes "μ1 ≠ μ2" "r1 > 0" "r2 > 0"
              "{z1, z2} ⊆ circle μ1 r1 ∩ circle μ2 r2" "z1 ≠ z2"
  shows "ang_circ_a z1 μ1 μ2 = ang_circ_a z2 μ1 μ2"
```

Ми смо доказали ову лему тек након примене „бгно" резоновања и померањем слике тако да центри два круга који се посматрају буду на x-оси.

У доказу смо идентификовали бројне дегенерисане случајеве који су морали да се анализирају одвојено. Прво смо морали да докажемо да кругови који се секу могу имати исти центар (тј. да $\mu_1{=}\mu_2$) само ако су једнаки и тада је оштар угао између њих једнак 0. Са друге стране, ако су оба центра колинеарна са пресечном тачком z (тј. ако важи `collinear` μ_1 μ_2 z), два круга се додирују (било споља или изнутра), и опет је оштар угао једнак 0.

Постојање круга R_i који је ортогоналан на јединичну кружницу и који има исту тангенту у датој тачки z као и дати круг са центром μ_i је дато следећом лемом (заправо у леми се даје центар μ_i' тог новог круга).

```
lemma
  assumes "⟨μᵢ - z, z⟩ ≠ 0"
          "μᵢ' = z + (1 - z*cnj z) * (μᵢ - z) / (2 * ⟨μᵢ - z, z⟩)"
  shows "collinear z μᵢ μᵢ'" "z ∈ ortho_unit_circ μᵢ'"
```

Аналитички израз је открио још неке дегенерисане случајеве. Бројилац може бити нула једино ако се кругови секу на јединичној кружници (тј. када је $z *$ `cnj` $z = 1$). У том случају, доказ из уџбеника се не може применити јер је $\mu_1' = \mu_2' = z$, и кругови R_1 и R_2 се не могу конструисати (они су празни кругови). Случај када је именилац једнак нули (било за μ_1' или μ_2') је такође дегенерисан. Ово се дешава када су вектори $\mu_i - z$ и z ортогонални. Геометријски, у том случају се круг R_i дегенерише у праву (што и није проблем у проширеној комплексној равни, али јесте проблем у поставци која важи у оригиналном доказу која се налази у обичној комлексној равни). Зато, овај специјалан случај мора да се анализира одвојено. Тако је наша формална анализа брзо показала да једноставно тврђење у Нидамовом уџбенику [130] да „за дату било коју тачку z која није на кругу инверзије K, постоји тачно један круг који је ортогоналан на K и пролази кроз z у било ком задатом правцу" није тачно у многим случајевима.

5.9 Закључци и даљи рад у формализацији геометрије комплексне равни

У овом раду смо показали неке елементе наше формализације геометрије проширене комплексне равни $\overline{\mathbb{C}}$ коришћењем комплексне пројективне равни, али и Риманове сфере. Формализовали смо аритметичке операције у $\overline{\mathbb{C}}$, размеру и дворазмеру, тетивну метрику у $\overline{\mathbb{C}}$, групу Мебијусових трансформација и њихово дејство на $\overline{\mathbb{C}}$, неке њене специјалне подгрупе (еуклидске сличности, ротације сфере, аутоморфизме диска), кругоправе и њихову везу са круговима и правама, тетивном метриком, Римановом сфером, јединственост кругоправи, дејство Мебијусових трансформација на кругоправе, типове и кардиналност скупа кругоправе, оријентисане кругоправе, однос између Мебијусових трансформација и оријентације, својство очувања угла након дејства Мебијусових трансформација итд. Наша тренутна теорија има око 12,000 линија *Isabelle/HOL* кода (сви докази су структурни и записани су у језику за доказе *Isabelle/Isar* и наши ранији покушаји су замењени крађим алгебарским доказима и нису укључени у финалну формализацију), око 125 дефиниција и око 800 лема.

Кључан корак у нашој формализацији је била одлука да се користи алгебарска репрезентација свих важних објеката (вектора хомогених координата, матрица за Мебијусове трансформације, хермитске матрице за кругоправе итд.). Иако ово није нов приступ (на пример, Швердфегерова класична књига [148] прати овај приступ прилично конзистентно), он ипак није тако уобичајен у литератури (и у материјалима курсева који се могу наћи на интернету). Уместо њега преовладао је геометријски приступ. Ми смо покушали да пратимо такву врсту геометријског резоновања у раној фази нашег рада на овој теми, али смо наишли на бројне потешкоће и нисмо имали много успеха. На основу овог искуства, закључујемо да увођење моћне технике линеарне алгебре омогућава значајно лакши рад на формализацији него што је то случај када се користи геометријско резоновање.

Може се дискутовати да ли у неким случајевима геометријски аргументи дају боље објашњење неких теорема, али када се посматра само доказивање тврђења, алгебраски приступ је јасно супериорнији. Ипак, да би имали везу са стандардним приступом у коме се користи геометријска интуиција увели смо неколико додатних дефиниција (које су више геометријске или више

алгебарске) и морали смо доказати да су ове дефиниције еквивалентне. На пример, када је дефиниција угла дата само коришћењем алгебарских операција на матрицама и њиховим детерминантама, својство очувања угла је било веома лако доказати, али због образовне сврхе ово постаје значајно једино када се та дефиниција споји са стандардном дефиницијом угла између кривих (тј. њихових тангентних вектора) — у супротном, формализација постаје игра са симболима који немају никакво значење.

Још један важан закључак до ког смо дошли је да у формалним документима треба што чешће избегавати анализу случајева и екстензије које омогућавају резоновање без анализе случајева треба што чешће користити (нпр. било је много боље користити хомогене координате уместо једне одвојене тачке бесконачно коју би морали засебно да анализирамо у сваком тврђењу или дефиницији; слично, било је много лакше радити са кругоправама него разликовати случај прави и кругова, итд.). Увођење два модела истог концепта (на пример, у нашем случају, хомогених координата и Риманове сфере) такође помаже, јер су неки докази лакши у једном моделу, а неки у другом.

У принципу наши докази нису дугачки (15-20 линија у просеку). Ипак, понекад је било потребно изводити веома досадне закључке, поготову када се пребацивало са реалних на комплексне бројеве и обратно (коришћењем функција за конверзију `Re` и `cor`). Ове конверзије се углавном и не појављују у неформалном тексту и добро би дошла нека аутоматизација оваквог закључивања. Аутоматизација система *Isabelle* је прилично моћна у резоновању једнакости у којима су обични комплексни бројеви и ту смо често користили метод `(simp add: field_simps)` (са неким мањим изузецима), али када су у питању неједнакости, аутоматизација није била добра и много тога смо морали да доказујемо ручно, корак по корак, а оваква тврђења се често сматрају веома тривијалним у неформалном тексту.

У нашем даљем раду планирамо да користимо ове резултате у формализацији нееуклидских геометрија и њихових модула (посебно, сферични модел елиптичке геометрије, Поенкареов диск модел и модел горње полуравни хиперболичке геометрије).

5.10 Формализација Поенкареовог диск модела

Циљ је доказати да Поенкареов диск модел представља модел свих аксиома Тарског са изузетком Еуклидове аксиоме која у овом моделу није тачна. Потребно је дефинисати основне појмове, тј. релацију *између* и растојање и показати да дефинисани појмови задовољавају нека својства. Нажалост, због разлога које ћемо касније изложити, нисмо успели да формално докажемо све аксиоме. Ипак, изложићемо нека интересантна својства и закључке до којих смо дошли.

Прво, дефинишемо тип података којим се представљају тачке Поенкареовог диск модела, односно тачке које припадају унутрашњости јединичног диска.

```
typedef unit_disc = "{z::complex_homo.  in_ocircline ounit_circle z}"
```

Специјална тачка која припада унутрашњости је и 0. Иако већ постоји дефинисана 0_h било је потребно дефинисани и 0_u, односно нулу која припада јединичном диску. Иако тривијално важи да 0_h припада јединичном диску, приликом дефинисања појмова или задавања лема ово није познато и систем може пријављивати грешку јер тип није одговарајући. Управо зато, потребно је дати још једну дефиницију за 0.

```
lift_definition zero_homo_unit :: unit_disc ("0_u") is zero_homo
```

Растојање

Растојање над тачкама јединичног диска се дефинише исто као и растојање над тачкама проширене комплексне равни.

```
lift_definition dist_poincare :: "unit_disc ⇒ unit_disc ⇒ real" is
    dist_homo
```

Релација *између*

Дефиниција релације *између* се ослања на већ дефинисани појам `cross_ratio` за који су већ доказана бројна својства. Релација *између* се прво дефинише над проширеном комплексном равни, а потом се подиже на тип `unit_disc`.

```
definition between where
   "between z1 z2 z3 ⟷ ((z1 = z2 ∧ z2 = z3) ∨
       (let CR = to_complex(cross_ratio z1 z2 z3 (inversion_homo z2))
        in
        is_real CR ∧ Re CR ≤ 0))"
lift_definition between_poincare ::
   "unit_disc ⇒ unit_disc ⇒ unit_disc ⇒ bool" is between
```

Као што се може видети у дефиницији разликујемо два случаја. Код Тарског, за тачке које су једнаке такође важи релација *између*, односно у моделу Тарског је допуштено да тачке буду једнаке. Како `cross_ratio` није дефинисан када су три тачке једнаке, то случај једнаких тачака одвајамо посебно.

Ако све четири тачке припадају једној кругоправој, онда ће двострука размера за те четири тачке бити реална. Неформално, тачке за које важи релација *између* припадају једној кругоправој и то не било каквој кругоправој, већ оној која је нормална на јединичну кружницу. Кругоправе нормалне на јединичну кружницу могу бити кругови нормални на јединичну кружницу или праве које пролазе кроз координатни почетак. Додатно, инверзија у односу на јединични круг било које од ове три тачке ће такође припадати кругоправој нормалној на јединичну кружницу. Зато, у двоструку размеру уврстимо три тачке и инверзију једне од њих и за ове четири тачке двострука размера мора бити реална.

Додатно, ако је дворазмера негативна онда је друга тачка између прве и треће, а у супротном није. Ако је дворазмера нула онда су две од три тачке једнаке и тада исто важи релација *између*.

Наравно, да би потврдили да је овако дефинисана релација заиста релација *између* потребно је доказати аксиоме Тарског за овај модел.

Мебијусове трансформације и релација *између*

Не посматрају се све Мебијусове трансформације већ само оне које сликају унутрашњост диска у унутрашњост диска. Раније смо видели да је $GU_{1,1}(\mathbb{C})$ група оних Мебијусових трансформација које фиксирају јединични круг, али да и ту постоје две групе трансформација, тј. оне трансформације које сликају унутрашњост круга у унутрашњост (и које су у овом контексту значајне)

и друга група трансформација које размењују унутрашњост и спољашњост диска.

Дефинисаћемо својство којим се описују оне Мебијусове трансформације које сликају унутрашњост диска у унутрашњост диска. И ова дефиниција се одвија у два корака, прво се дефинише над матрицама, а потом се подигне на тип `moebius`.

```
definition Unitary11_gen_direct_rep where
   "Unitary11_gen_direct_rep M ⟷
   (let (A, B, C, D) = ⌊M⌋_M
   in unitary11_gen (A, B, C, D) ∧ (B = 0 ∨ Re ((A*D)/(B*C)) > 1))"

lift_definition Unitary11_gen_direct :: "moebius ⇒ bool" is
   Unitary11_gen_direct_rep
```

Може се доказати следеће тврђење

```
lemma
   "moebius_ocircline M ounit_circle = ounit_circle ⟷
   Unitary11_gen_direct M"
```

односно, унутрашњост диска је очувана ако и само ако Мебијусова трансформација задовољава својство `Unitary11_gen_direct`.

Сада се може дефинисати тип, тј. нова група Мебијусових трансформација које чувају унутрашњост диска

```
typedef moebius_unitary = "{M::moebius. Unitary11_gen_direct M}"
```

Слично као и раније, Мебијусове трансформације ове групе су дате као дејство над тачкама јединичног диска. Ипак, за ову дефиницију се може искористити дефиниција Мебијусових трансформација над тачкама проширене комплексне равни, тј. потребно је подићи ту дефиницију за тип `mobius_unitary` и `unit_disc`.

```
lift_definition moebius_pt_poincare ::
   "moebius_unitary ⇒ unit_disc ⇒ unit_disc" is moebius_pt
```

Ова дефиниција ствара обавезу да се докаже

```
∀ M z.
    ⟦ Unitary11_gen_direct M; in_ocircline ounit_circle z ⟧
    ⟹ in_ocircline ounit_circle (moebius_pt M z)
```

што се лако доказује у неколико корака коришћењем горе дате леме.

Поред ових дефиниција интересантно је дефинисати и инверзну трансформацију.

```
lift_definition moebius_inv_poincare::
  "moebius_unitary ⇒ moebius_unitary" is moebius_inv
```

И ова дефиниција ствара обавезу да се докаже да је инверзна трансформација такође Unitary11_gen_direct што се доказује коришћењем једноставних алгебарских трансформација над матрицама.

Веома важно тврђење које смо доказали је да Мебијусове трансформације које чувају унутрашњост диска, такође чувају и релацију *између*. Ово тврђење се веома лако доказује јер смо раније већ доказали да Мебијусове трансформације чувају дворазмеру. Ипак, било је потребно доказати и да је очувана инверзија у односу на јединични круг. У општем случају инверзија није очувана, али за Мебијусове трансформације које чувају унутрашњост јединичног диска, јесте.

```
lemma
  assumes "Unitary11_gen_direct M"
  shows "moebius_pt M (inversion_homo z) =
         inversion_homo (moebius_pt M z₂)"


lemma
  assumes "z₁' = moebius_pt_poincare M z₁"
          "z₂' = moebius_pt_poincare M z₂"
          "z₃' = moebius_pt_poincare M z₃"
          "between_poincare z₁ z₂ z₃"
  shows "between_poincare z₁' z₂' z₃'"
```

Поред ових доказано је још пуно помоћних тврђења, а издвојићемо неколико интересантнијих. Једна од често коришћених чињеница у доказима је да инверзна слика тачке јединичног диска не припада јединичном диску.

```
lemma
  assumes "x ∈ unit_disc"
  shows "inversion_homo x ∉ unit_disc"
```

Такође, тачке које се налазе у јединичном диску су по модулу мање од 1.

```
lemma
  assumes "in_ocircline ounit_circle z"
  shows "cmod (to_complex z) < 1"
```

Важно тврђење је да свака Мебијусова трансформација која је композиција ротације и Блашке фактора чији параметар је по модулу мањи од 1 је трансформација која слика унутрашњост диска у унутрашњост диска.

```
lemma
  assumes "cmod a < 1" "a * cnj a ≠ 1" "a ≠ 0"
          "M = rotation_moebius φ + blaschke a"
  shows "Unitary11_gen_direct M"
```

Прво се докаже да се било које две тачке могу сликати у 0_u и у неку тачку на реалној оси, што је тврђење дато у следећој леми.

```
lemma
  "∃ a M. 0u = moebius_pt_poincare M z1 ∧
           is_real (to_complex (Rep_unit_disc a)) ∧
           a = moebius_pt_poincare M z2"
```

Доказ овог тврђења се састоји из два корака. Прво се прва тачка слика у 0_u трансформацијом M' `= blaschke (to_complex` z_1), а потом се врши ротација за угао који одговара тачки која се добила пресликавањем тачке z_2 трансформацијом M'. Како је ротација око координатног почетка, то се 0_u слика у 0_u, а друга тачка се ротацијом слика на реалну осу и тиме се управо и добијају жељене слике тачака. У доказу се користи раније доказана чињеница да је композиција ротације и Блашке фактора чији је параметар имао модуо мањи од 1 Мебијусова трансформација која чува унутрашњост диска. Иако је идеја доказа једноставна, постоји неколико случајева које треба размотрити (ако су тачке једнаке или ако је већ нека тачка једнака 0_h), а и у доказу

постоји много ситних корака које је требало доказати. Зато је доказ готово 300 линија дугачак.

Потом се може доказати да ако за три тачке важи релација *између*, онда се оне могу сликати на реалну осу (а једна од њих у 0). У овом доказу се користи претходно тврђење и чињеница да ако је дворазмера реална, ако су њена три параметра реална, онда и четврти параметар мора бити реалан.

```
lemma
   assumes "between_poincare z₁ z₂ z₃"
   shows "∃ a b M. 0ᵤ = moebius_pt_poincare M z₁ ∧
         is_real (to_complex (Rep_unit_disc a)) ∧
         a = moebius_pt_poincare M z₂ ∧
         is_real (to_complex (Rep_unit_disc b)) ∧
         b = moebius_pt_poincare M z₃"
```

Коришћењем свих претходних тврђења може се користити резоновање „без губитка на општости". Наиме, тврђење се може доказати на реалној оси на којој је лакше доказати да неко тврђење важи, а онда се може уопштити да важи за било које три тачке за које важи релација *између* у Поенкареовом диск моделу.

Аксиоме подударности

Аксиоме подударности се тривијално доказују јер су својства растојања већ раније доказана и само је потребно позвати се на та својства.

```
lemma ax₁:
   "dist_poincare z₁ z₂ = dist_poincare z₂ z₁"
lemma ax₂:
   assumes "dist_poincare x y = dist_poincare z z"
   shows "x = y"
lemma ax₃:
   assumes "dist_poincare x y = dist_poincare z u"
           "dist_poincare x y = dist_poincare v w"
   shows "dist_poincare z u = dist_poincare v w"
```

Аксиоме релације *између*

Две аксиоме ове групе се тривијално доказују. Аксиома идентитета се доказује контрадикцијом. Аксиома горње димензије тврди да постоје три тачке за које не важи релација *између*. Доказује се тако што се одаберу такве три тачке, на пример, 0_u, $1/2$ и $\mathtt{ii}/2$, докаже се да све три тачке заиста припадају Поенкареовом диск моделу, али да дворазмера није реална, па самим тим и релација *између* не важи.

```
lemma ax4:
   assumes "between_poincare x y x"
   shows "x = y"
lemma ax6:
   "∃ a b c. ¬ between_poincare a b c ∧ ¬ between_poincare b c a
   ∧ ¬ between_poincare c a b"
```

Овој групи аксиома припада аксиома непрекидности.

```
lemma ax7:
   assumes "∃ a. ∀ x. ∀ y. ϕ x ∧ ψ y ⟶ between_poincare a x y"
   shows "∃ b. ∀ x. ∀ y. ϕ x ∧ ψ y ⟶ between_poincare x b y"
```

Да би доказали ово тврђење било је потребно доказати пуно помоћних лема. Доказ се започиње једноставним испитивањем случајева у којима тврђење тривијално важи. Први случај је да не постоје ни x, ни y такви да важи $\phi\ x$ и $\psi\ y$. Следећи случај је да не постоји x такво да важи $\phi\ x$. Трећи случај је да не постоји y такво да важи $\psi\ y$. И последњи тривијалан случај је да постоји b такво да важи $\phi\ b$ и $\psi\ b$.

Доказивање општег случаја се састоји из неколико корака. Прво се све тачке сликају на реалну осу на којој је тврђење лакше доказати. Оправданост овог пресликавања смо видели раније. Потом је потребно доказати једно тврђење за релацију *између* које важи на реалној оси. То тврђење тврди да тачке које су у релацији *између* задовољавају неки поредак.

```
lemma
   assumes "is_real (to_complex (Rep_unit_disc z))"
           "is_real (to_complex (Rep_unit_disc u))"
           "is_real (to_complex (Rep_unit_disc v))"
```

```
        "rz = Re (to_complex (Rep_unit_disc z))"
        "ru = Re (to_complex (Rep_unit_disc u))"
        "rv = Re (to_complex (Rep_unit_disc v))"
  shows "between_poincare z u v ⟷
        (rz ≤ ru ∧ ru ≤ rv) ∨
        (rz ≥ ru ∧ ru ≥ rv)"
```

Потом доказујемо аксиому непрекидности за реалне бројеве:

```
lemma
  assumes "∀ x::real. ∀ y::real. φ x ∧ ψ y ⟶ x < y"
          "∃ x. φ x" "∃ y. ψ y"
  shows "∃ b. ∀ x. ∀ y. φ x ∧ ψ y ⟶ (x ≤ b ∧ b ≤ y)"
```

Ово тврђење се једноставно доказује коришћењем својства супремума. Наиме, посматра се скуп `P = "{x::real. φ x}"`. Докаже се да супремум овог скупа управо испуњава тражено тврђење.

Комбинујући последње две леме, доказује се и тврђење аксиоме.

Преостале аксиоме Тарског нисмо успели да докажемо. Узмимо Пашову аксиому у разматрање.

```
lemma ax5:
  assumes "between_poincare x u z"
          "between_poincare y v z"
  shows "∃ a. between_poincare u a y ∧ between_poincare v a x"
```

Проблем пресека кругоправих Да би доказали ово тврђење потребно је одредити x које испуњава тражена својства. То значи да треба одредити x као пресек две кругоправе нормалне на јединичну кружницу. Прва кругоправа садржи тачке u и y, а друга кругоправа садржи тачке v и x. То значи да треба одредити пресек два круга.

Коришћењем раније показаних својства лако се може одредити хермитска матрица H_1 кругоправе која садржи u и v. То је матрица која је нормална на јединични круг, па је облика $\begin{pmatrix} A & B \\ \bar{B} & A \end{pmatrix}$. Поред овога, још важи `on_circline_rep` H_1 u (односно `quad_form` $\lfloor H_1 \rfloor_H$ $\lfloor u \rfloor_{C2} = 0$, тј. $\bar{u'} \cdot H_1 \cdot u' = 0$, при

чему u' представља хомогене координате тачке u), и слично и за другу тачку, односно `on_circline_rep` H_1 v (тј. `quad_form` $\lfloor H_1 \rfloor_H$ $\lfloor v \rfloor_{C2} = 0$). Развијањем се добије систем једначина по коефицијентима кругоправе H_1 који се може лако решити. Слично је и у другом случају, односно приликом одређивања хермитске матрице кругоправе H_2 којој припадају тачке v и x. Када добијемо ове две матрице, потребно је одредити тачку која припада обема, односно тачку која задовољава услове `quad_form` $\lfloor H_1 \rfloor_H$ $\lfloor q \rfloor_{C2}$ и `quad_form` $\lfloor H_2 \rfloor_H$ $\lfloor q \rfloor_{C2}$, односно

$$\bar{q'} \cdot H_1 \cdot q' = 0$$

$$\bar{q'} \cdot H_2 \cdot q' = 0$$

при чему q' представља хомогене координате пресечне тачке и то је непозната променљива. Идеално решење би било када би овај систем једначина могао да се реши коришћењем матричних трансформација, али ми нисмо успели да нађемо такво решење. Расписивањем овог система по координатама матрице и хомогених вектора добија се квадратна једначина и резултат пресека ће бити корен неког великог израза.

Претходни систем је могуће за нијансу упростити тако што би се упростила матрица која представља кругоправу. Наиме, посматрајмо прву кругоправу која пролази кроз тачке v и x и чија хермитска матрица је H_1. Коришћењем Мебијусових трансформација које чувају јединични диск (било је речи раније 5.10) могуће је сликати тачку v у координатни почетак, а тачку x на x–осу. Наиме, посматрајмо матрицу M `=` `rotation_moebius` ϕ `+` `blaschke` v при чему је ϕ `=` `arg ((blaschke` v`)` x`)`. Ова матрица представља Мебијусову трансформацију која слика кругоправу H_1 на x–осу, а тачку v на $(0, 0)$. Ову матрицу можемо применити на обе кругоправе, тј. $H_1' =$ `mat_adj` $M \cdot H_1 \cdot M$ и $H_2' =$ `mat_adj` $M \cdot H_2 \cdot M$. Тада се систем једначина своди на

$$\bar{q'} \cdot H_1' \cdot q' = 0$$

$$\bar{q'} \cdot H_2' \cdot q' = 0$$

при чему знамо да је H_1' једнака x–оси, односно да је облика $(0, i, -i, 0)$. Ипак, друга кругоправа може бити било шта (круг или права), што значи да тражимо пресек праве и круга, што је и даље квадратна једначина и решење ће поново бити комплексан израз.

Алгебарским трансформацијама може се одредити корени израз који представља пресек кругова (и како су кругови, у општем случају биће два пресека).

Ипак, са добијеним комплексним изразом је тешко резоновати. Први изазов је одредити који од два пресека заиста припада диску и јесте тражени пресек. Потом је потребно комплексан израз који садржи квадратни корен уврстити у једначину дворазмере, и проверити да ли је добијени израз реалан и негативан. Ово се показало као веома тежак задатак. Наиме, није могуће лако се ослободити корена, а изрази који се добијају су веома комплексни и тешко је радити са неједнакостима и доћи до жељених закључака. Зато, нажалост, овај доказ остаје незавршен.

Исти су проблеми и у другим доказима. Такође је потребно пронаћи пресеке кругоправих, а онда уврстити то у вектор или матрицу и резоновати са тим комплексним изразима.

У многим уџбеницима [130, 148] смо наишли на тврђење да се тривијално доказује да је Поенкареов диск модел модел аксиома Тарског изузимајући Еуклидову аксиому. Ипак, ни у једном уџбенику, за сада, нисмо пронашли доказ овог тврђења. Нама није успело да самостално доказ и довршимо, а сматрамо да сам доказ није тривијалан.

Закључци и даљи рад

Иако је тврђење да је Поенкареов диск модел модел геометрије Лобачевског део математичког фолклора, показало се да је ово тврђење јако тешко формализовати. Један од првих изазова је била дефиниција релације *између*. Релација се може дефинисати на два начина, пратећи геометријски приступ или пратећи алгебарски приступ. Као и раније, алгебарски приступ се показао далеко супериорнији и зато смо одабрали да ову релацију дефинишемо коришћењем дворазмере. У оквиру формализације комплексне равни доказана су бројна својства дворазмере, па је доказивање многих својстава релације *између* било веома једноставно. Показано је да шест аксиома Тарског важе у Поенкареовом диск моделу и да Еуклидова аксиома паралелности не важи. Ипак, доказивање да остале аксиоме важе је било проблематично јер за доказ тих аксиома је потребно одредити пресек кругоправих што рачун и доказе чини значајно комплекснијим. То је разлог зашто ова формализација није завршена до краја. И поред овог проблема, бројна својства релације *између* и релације *подударно* у оквиру Поенкареовог диск модела су показана, а показано је и да једна група Мебијусових трансформација чува ове две релације. То се може користити у даљим формализацијама.

Глава 6

Формална анализа алгебарских метода и њихове примене на проблеме у стереометрији

6.1 Увод

Објекти и релације еуклидске геометрије могу бити описани коришћењем полинома. Додатно, свака геометријска конструкција може бити изражена скупом полинома, а многа геометријска тврђења могу бити доказана коришћењем алгебарских метода као што су Гребнерове базе или Вуов метод над скупом полинома. Описаћемо имплементацију алгоритма у систему *Isabelle/HOL* који као улазне податке прихвата термове који описују геометријску конструкцију и враћа одговарајући скуп полинома. Даљи циљ је примена метода Гребнерових база у оквиру система *Isabelle/HOL* над генерисаним полиномима у намери да се докаже исправност тврђења.

Главна идеја је да се повеже аутоматско и формално доказивање у геометрији. И даље не постоји јединствен, нити верификован алгоритам који трансформише геометријску конструкцију у скуп полинома. Обично, превођење у полиноме се ради ад-хок методама и не постоји формална веза између добијених полинома и датих геометријских објеката. Са формално верификованим методом превођења овај проблем би био решен и у оквиру ове тезе биће представљени кораци у том правцу.

6.2 Алгебарски методи у геометрији

Превођење геометријских тврђења у алгебарску форму

Алгебарски методи се користе у аутоматском доказивању у геометрији за теореме конструктивног типа, тј. тврђења о геометријским објектима које су добијене током геометријске конструкције. Уводе (симболичке) координате за геометријске објекте (тачке, и понекад праве) који се јављају у конструкцији и геометријске конструкције и тврђења изражавају као алгебарске једначине у којима се јављају уведене координате. Тиме се уместо разматрања еуклидске геометрије прелази на разматрање у Декартовој координатној равни, тј. једном моделу еуклидске геометрије. Након увођења координата користе се алгебарске технике да докажу да тврђење следи из конструкције.

Пре него што је могуће применити алгебарске методе, геометријско тврђење мора бити записано у алгебарској форми, као скуп полиномијалних једнакости (односно полинома), при чему се претпоставља да су све полиномијалне једнакости облика $f(x) = 0$. Процедуре за алгебризацију најчешће уводе нове симболичке промењљиве за координате тачке и уводе полиномијане једнакости који карактеришу сваки конструктивни корак и свако тврђење које је потребно доказати. Иако је могуће и за праве које се појављују у конструкцији увести непознате коефицијенте као симболичке промењљиве, уобичајено је да се избегава такав приступ и користе се само тачке (а праве су имплицитно задате). Свака конструкција почиње скупом слободних тачака и током конструкције се уводе зависне тачке. У неким случајевима је могуће зависне тачке бирати са неким степеном слободе (на пример, избор произвољне тачке на датој правој).

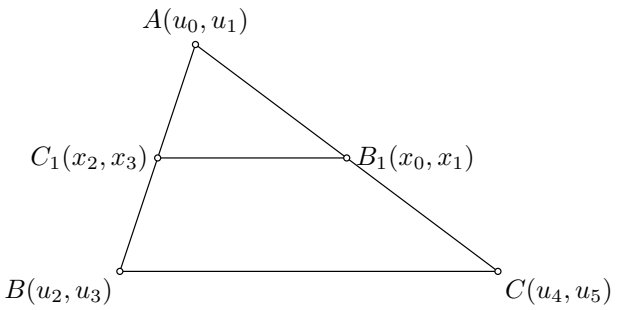
Свака тачка добија координате које су репрезентоване симболичким промењљивима. Најчешће се слободне промењљиве означавају са u_i ($i = 0, 1, 2, \ldots$), а зависне промењљиве се означавају са x_i ($i = 0, 1, 2, \ldots$). Ако је тачка зависна, онда ће све њене координате бити зависне промењљиве. Ако је тачка слободна, онда су њене координате такође слободне. Ако је тачка зависна, али са неким степеном слободе, онда једна координата у дводимензионалном случају или две координате у тродимензионом случају координате могу бити слободне. Ипак, ако постоји избор коју координату изабрати за зависну, а коју за слободну, то питање није тривијално и захтева посебну пажњу. На пример, у Декартовој координатној равни, ако је тачка A произвољна тачка

праве l, једна од њених координата може бити слободна, а једна зависна и може бити израчуната на основу ограничења која важе за праву. Ипак, ако је права l паралелна са x-осом, онда y координата тачке A не може бити слободна. Слично, ако је l паралелна са y-осом онда x координата тачке A не може бити слободна.

Геометријска ограничења која важе за тачке могу се формулисати у виду алгебарских ограничења над координатама тачке (тј. као полиномијалне једначине над уведеним симболичким координатама). На пример, претпоставимо да су симболичке координате за тачку A (a^x, a^y), за тачку B (b^x, b^y), а за тачку C (c^x, c^y). Чињеница да је тачка A средишња тачка интервала BC одговара алгебарским условима $2{\cdot}a^x = b^x{+}c^x$ и $2{\cdot}a^y = b^y{+}c^y$. Услов да су тачке A, B и C колинеарне одговара алгебарском услову $(a^x{-}b^x)(b^y{-}c^y) = (a^y{-}b^y)(b^x{-}c^x)$. Слични услови се могу формулисати и за друге основне геометријске релације (паралелне праве, нормалне праве, симетрала дужи итд.).

Примери

***Пример* 6.2.1.** *Дат је троугао ABC. Нека је B_1 средишња тачка дужи AC, C_1 нека је средишња тачка интервала AB. Доказати да је средња линија B_1C_1 паралелна страници троугла BC.*



Слика 6.1: Теорема о средњој линији троугла

Бирамо координатни систем тако да темена фигуре буду у канонском положају, те је стога први корак поставити координате за свако теме троугла. Тачке A, B и C су слободне тачке, те им додељујемо координате $A(u_0, u_1)$, $B(u_2, u_3)$ и $C(u_4, u_5)$. Тачке B_1 и C_1 су зависне, те им додељујемо

координате $B_1(x_0, x_1)$ *и* $C_1(x_2, x_3)$. *Како је* B_1 *средишња тачка дужи* AC, *важи*

$$2 \cdot x_0 - u_0 - u_4 = 0$$

$$2 \cdot x_1 - u_1 - u_5 = 0$$

Полином са леве стране прве једначине означићемо са f_1, *а полином са леве стране друге једначине означићемо са* f_2.

Тачка C_1 *је средишња тачка дужи* AB, *па важи:*

$$2 \cdot x_2 - u_0 - u_2 = 0$$

$$2 \cdot x_3 - u_1 - u_3 = 0$$

Полином са леве стране прве једначине означићемо са f_3, *а полином са леве стране друге једначине означићемо са* f_4.

Са ове четири једначине дат је опис конструкције. Кажемо да полиноми f_1, f_2, f_3 *и* f_4 *припадају скупу–конструкције.*

Потребно је доказати да је BC *паралелно са* B_1C_1, *односно, записано полиномима, потребно је да важи:*

$$(x_2 - x_0)(u_5 - u_3) - (x_3 - x_1)(u_4 - u_2) = 0$$

Полином са леве стране једначине означићемо са g *и то је полином тврђења (припада скупу–тврђења).*

Потребно је доказати да свака n*-торка која анулира полиноме конструкције такође анулира и полиноме тврђења, тј. потребно је доказати да*

$$\begin{aligned}\forall\ u_0\ u_1\ u_2\ u_3\ u_4\ u_5\ x_0\ x_1\ x_2\ \ x_3 \in \mathbb{R}.\ & 2 \cdot x_0 - u_0 - u_4 = 0 \wedge 2 \cdot x_1 - u_1 - u_5 = 0 \\ & \wedge\ 2 \cdot x_2 - u_0 - u_2 = 0 \wedge 2 \cdot x_3 - u_1 - u_3 = 0 \\ & \Longrightarrow (x_2 - x_0)(u_5 - u_3) - (x_3 - x_1)(u_4 - u_2) = 0\end{aligned}$$

Приметимо да у претходном примеру услов да је ABC троугао није преведен у услов да су тачке A, B и C међусобно различите. Додатно, услов да је B_1C_1 паралелно са BC је дато условом једначином $(x_2 - x_0)(u_5 - u_3) - (x_3 - x_1)(u_4 - u_2) = 0$. Са друге стране, ова алгебарска једначина је једнака једном слабијем услову, наиме услову: $B \equiv C$ или $B_1 \equiv C_1$ или B_1C_1 паралелно са BC. Преведено у геометријски запис, тврђење које је доказано алгебарском методом је следеће:

Нека је B_1 средишња тачка дужи AC и C_1 нека је средишња тачка интервала AB. Онда је B_1C_1 паралелно са BC или је B идентично тачки C или је тачка B_1 идентична тачки C_1.

Како $B_1 \not\equiv C_1$ следи из $B \not\equiv C$, претходно тврђење је еквивалентно са:

Нека су B и C две различите тачке. Нека је B_1 средишња тачка дужи AC и C_1 нека је средишња тачка дужи AB. Онда је дуж B_1C_1 паралелна са BC.

Овај пример показује да превођење геометријског тврђења у алгебарски запис и обратно захтева да се обрати пажња на многе детаље. У већини система, хипотеза облика $AB||CD$ се обично записује једначином облика $(b^x - a^x)(d^y - c^y) = (d^x - c^x)(b^y - a^y)$, чак се ова једначина користи као дефиниција за $AB||CD$. Ипак, овакав приступ раскида везу са синтетичком геометријом.

У зависности од конструкције и тврђења може бити много полинома који припадају скупу–конструкције или скупу–тврђења. Ова два скупа су веома важна за метод Гребнерових база (или за Вуов метод) и касније биће детаљније појашњена њихова улога.

Може се доказати да је већина геометријских својстава инваријантна у односу на изометријске трансформације [53, 78]. Ако су P_1 и P_2 две слободне тачке, увек постоји изометрија (прецизније, композиција осних рефлексија) која слика P_1 у тачку $(0, 0)$ (тј. у координатни почетак), а тачку P_2 у тачку на x-оси (или у тачку на y-оси). Зато се без губитка на општости може претпоставити да слободна тачка има координате $(0, 0)$, док друга тачка има координате $(u_0, 0)$ или $(0, u_0)$ (иако су оба избора коректна, у неким случајевима избор може утицати на ефикасност, или, у случају једноставног Вуовог метода, избор може утицати на могућност доказивања). Примена овог закључка може значајно утицати на обим посла који има алгебарска метода. Уз то, постоје хеуристике (са циљем да побољшају ефикасност) за избор која од слободних тачака је најпогоднија за ове специјалне координате.

***Пример* 6.2.2.** *Без губитка на општости у Примеру 6.2.1, тачкама B C могу бити додељене координате $B(0, 0)$ и $C(u_4, 0)$. Тада се алгебарско тврђење које треба доказати своди на:*

$$\begin{aligned}\forall\ u_0\ u_1\ u_4\ x_0\ x_1\ x_2\ x_3 \in \mathbb{R}.\ \ & 2 \cdot x_0 - u_0 - u_4 = 0 \wedge 2 \cdot x_1 - u_1 = 0 \\ & \wedge\ 2 \cdot x_2 - u_0 = 0 \wedge 2 \cdot x_3 - u_1 = 0 \\ & \Longrightarrow (x_2 - x_0) \cdot 0 - (x_3 - x_1) \cdot u_4 = 0\end{aligned}$$

што тривијално следи (x_3—$x_1 = 0$ следи из $2 \cdot x_1 - u_1 = 0$ и $2 \cdot x_3 - u_1 = 0$).

Алгебарски алгоритми

Када се геометријско тврђење преведе у алгебарску форму, могуће је применити алгебарски метод за доказивање теорема. Алгебарски доказивачи теорема користе специфичан алгоритам над системом полинома. Ако су $f_1, \ldots, f_k$ полиноми скупа–конструкције, а $g_1, \ldots, g_l$ полиноми који су добијени из тврђења, онда се доказивање теореме своди на проверу да ли је за свако g_i испуњено:

$$\forall v_1, \ldots, v_n \in \mathbb{R} \bigwedge_{i=1}^{k} f_i(v_1, \ldots, v_n) = 0 \Longrightarrow g_i(v_1, \ldots, v_n) = 0$$

Тарски је приметио да се коришћењем елиминације квантификатора за реалне бројеве може доћи до доказа. Али у пракси, тешко је доказати нетривијална математичка тврђења на овај начин. Зато се примењује другачији приступ. Главна идеја, коју је предложио Wu 1978. године је да велики број геометријских тврђења, које се формулишу као универзална алгебарска тврђења у терминима координата, су такође тачна и за комплексне вредности координата. Уместо проверавања полинома над реалним бројевима, користи се поље комплексних бројева и посматра се следећа претпоставка [1]:

$$\forall v_1, \ldots, v_n \in \mathbb{C} \bigwedge_{i=1}^{k} f_i(v_1, \ldots, v_n) = 0 \Longrightarrow g_i(v_1, \ldots, v_n) = 0 \tag{6.1}$$

Ово је тачно ако g припада идеалу $I = \langle f_1, \ldots f_k \rangle$ који је генерисан над полиномима f_i $(i = 1, \ldots k)$, тј. када постоји цео број r и полиноми $h_1, \ldots h_l$ такви да $g_i^r = \sum_{i=1}^{k} h_i f_i$. Хилбертова (Nullstellensatz) теорема тврди да ако је поље алгебарски затворено (а $\mathbb{C}$ јесте) онда је обрнуто такође тачно.

Два најзначајнија алгебарска метода користе врсту еуклидског дељења да провере исправност претпоставке дате у 6.1. Бухбергеров алгоритам трансформише полазни скуп у *Гребнерову базу* у којој алгоритам дељења се може ефикасно употребити, а Вуов метод користи *псеудо–дељење* које на неки начин имитира еуклидско дељење.

[1] Наравно, постоји проблем некомплетности метода (у односу на геометрију) јер у неким случајевима тврђење важи у $\mathbb{R}$, али метод не успева да докаже због контрапримера који важе у $\mathbb{C}$.

Вуов метод

Главна операција над полиномима у Вуовом методу је псеудо–дељење које када се примени на два полинома $p(v_1, \ldots, v_n)$ и $q(v_1, \ldots, v_n)$ производи декомпозицију

$$c_m \ p = tq + r$$

при чему је $c(v_1, \ldots, v_{n-1})$ водећи коефицијент у полиному q уз променљиву v_n, m је број ненула коефицијената полинома p, $t(v_1, \ldots, v_n)$ је псеудо–количник, $r(v_1, \ldots, v_n)$ је псеудо–остатак, степен v_n у r је мањи него у q. Како важи $r = c_m \ p - tq$, јасно је да r припада идеалу генерисаном над полиномима p и q.

Први корак (једноставног) Вуовог метода [28] користи псеудо–дељење да трансформише конструисани систем полинома $(\bigwedge_{i=1}^{k} f_i)$ у троугаону форму, тј. у систем једначина у коме свака наредна једначина у систему уводи тачно једну нову зависну променљиву. Након тога, коначни остатак се рачуна псеудо–дељењем полинома тврђења (g_i) са сваким полиномом троугаоног система.

Вуов метод у свом најједноставнијем облику омогућава израчунавање полинома $c, h_1, \ldots, h_k$ и r таквих да важи

$$cg_i = \sum_{i=1}^{k} h_i f_i + r$$

Ако је коначни остатак r једнак нули, онда се сматра да је претпоставка доказана. Једноставан Вуов метод није комплетан (у алгебарском смислу). Комплекснија и комплетна верзија метода користи растуће ланце који се разматрају у оквиру Рит-Вуовог принципа.

Припадност идеалу, Гребнерове базе, пример примене Гребнеровог метода на конкретан проблем

У овом одељку су дате основне дефиниције и теореме које представљају математичку основу за овај рад. Прва дефиниција је дефиниција проблема који треба решити. Друга дефиниција дефинише Гребнерове базе, алатку која даје одговор на проблем припадности идеалу. Коначно, дата је и теорема која спаја ове две дефиниције.

Дефиниција 6.2.1 (Припадност идеалу)**.** За дате $f, f_1, \ldots, f_k \in K[X_1, \ldots, X_n]$ где су $f, f_1, \ldots, f_k$ полиноми, а $K[X_1, \ldots, X_n]$ је прстен полинома над K, и $\langle f_1, \ldots f_k\rangle$ је идеал генерисан са $f_1, \ldots f_k$, тада је проблем припадности идеалу проблем одређивања да ли је $f \in \langle f_1, \ldots f_k\rangle$ задовољено.

Дефиниција 6.2.2. Нека $I = \langle f_1, \ldots, f_n\rangle$ је идеал генерисан коначним скупом полинома. G је **Гребенерова база** идеала I ако и само ако је дељење полинома са више променљивих (означено са $\rightarrow_G$) било ког полинома у идеалу I са G даје 0.

***Теорема* 6.2.1.** *Проблем $f \in I$ је еквивалентан $f \xrightarrow{*}_G 0$.*

Оно што је заправо речено овом теоремом је да ако постоји скуп полинома конструкције и ако су сви полиноми једнаки нули, онда је могуће одредити Гребнерову базу за овај скуп и ако је могуће доказати (коришћењем полинома Гребнерове базе) да су сви полиноми тврђења једнаки нули онда је геометријско тврђење тачно. За израчунавање Гребнерове базе користи се *Бухбергеров алгоритам* који се лако може имплементирати. Данас постоје многе хеуристике које омогућавају да израчунавања буду бржа, али ми ћемо га представити у његовој основном облику:

Дефиниција 6.2.3. S-полином над полиномима f_i и f_j, означен са $S(f_i, f_j)$ се израчунава на следећи начин:

1) $m = GCD(H(f_i), H(f_j))$

2) $m = m_i * H(f_i)$ при чему је $H(f_i)$ водећи моном у f_i

3) $m = m_j * H(f_j)$ при чему је $H(f_j)$ водећи моном у f_j

4) $S(f_i, f_j) = m_i * f_i - m_j * f_j$

За скуп полинома $\{f_i, ..., f_j\}$ Бухбергеров алгоритам се састоји из следећих корака:

1) S-полином се одређује за свака два полинома из скупа чији водећи мономи нису узајамно прости и новодобијени полиноми се додају у скуп.

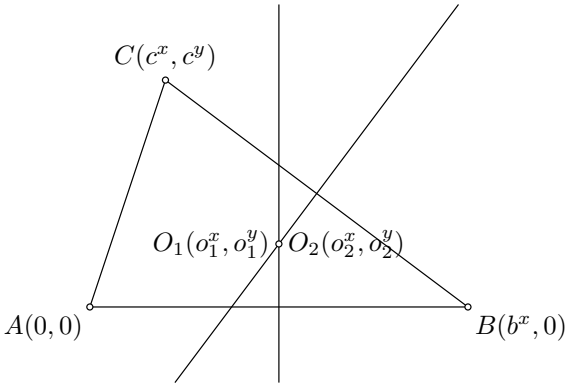
2) Понавља се први корак док год има полинома који могу бити додати.

Метод Гребнерових база је већ имплементиран у систему *Isabelle/HOL* и може се користити на следећи начин:

```
lemma "⟦−2 · u1 + x1 + x2 = (0::real); −2 · v1 + y1 + y2 = 0⟧ ⟹
    x1 · v1 − x1 · y2 + u1 · y2 − v1 · x2 − y1 · u1 + y1 · x2 = 0"
by algebra
```

при чему је метод Гребнерових база позван са **by algebra**.

***Пример* 6.2.3.** *Доказати да се симетрале страница троугла секу у једној тачки.*



Слика 6.2: Теорема о симеtралама страница троугла

Бирамо координатни систем тако да темена фигуре буду у канонском положају. То значи да је први корак поставити координате троугла тако што поставимо тачку A у координатни почетак, односно $A = (0, 0)$. Потом, можемо одабрати да су координате тачке $B = (b^x, 0)$, а тачке $C = (c^x, c^y)$ (тврђење је потребно доказати за било које координате тачака, али се лако може доказати да је могуће било које координате транслирати у ове које су специјално одабране и олакшавају даља израчунавања). Тачке A, B и C су слободне и њихове координате би могли означити са u_i ($i = 1, 2, 3$), али да би лакше пратили ознаке у овом примеру, означили смо их другачије. Сада се геометријска конструкција преводи у скуп полинома.

Прво записујемо хипотезу да се симетрале страница AB и BC секу у тачки $O_1 = (o_1^x, o_1^y)$. Како су симетрале потпуно одређене тачкама A, B, C и O_1, добијају се следеће једначине:

$$f_1 : \qquad o_1^x - \tfrac{b^x}{2} = 0$$

$$f_2 : \quad \tfrac{b^x - c^x}{c^y} \cdot o_1^x - o_1^y + \tfrac{c^{y\,2} - b^{x\,2} + c^{x\,2}}{2 \cdot c^y} = 0$$

Сада посматрамо другу хипотезу – симетрале страница AB и AC се секу у тачки $O_2 = (o_2^x, o_2^y)$. Добијамо две нове једначине:

$$f_1' : \qquad o_2^x - \tfrac{b^x}{2} = 0$$

$$f_3 : \quad \tfrac{c^x}{c^y} \cdot o_2^x + o_2^y - \tfrac{c^y}{2} - \tfrac{c^{x\,2}}{2 \cdot c^y} = 0$$

Значи, имамо скуп полинома:

$$G = \{f_1, f_2, f_1', f_3\}$$

Овим полиномима је заправо описана конструкција. Сада је потребно доказати да важи $O_1 = O_2$, тј. $(o_1^x, o_1^y) = (o_2^x, o_2^y)$. То значи да је потребно одредити Гребнерову базу G' скупа G и циљ је доказати $o_1^x - o_2^x \to_{G'} = 0$ и $o_1^y - o_2^y \to_{G'} = 0$. Са ова два полинома (лева страна датих једначина) дати су полиноми тврђења.

У намери да се израчуна Гребнерова база скупа G користи се Бухбергеров алгоритам и добијени резултат је:

$$\begin{aligned} G' = \{f_1, f_2, f_3, f_4, f_5, f_6\} = \{ & o_1^x - \frac{b^x}{2}, \quad o_2^x - \frac{b^x}{2}, \\ & \frac{b^x - c^x}{c^y} \cdot o_1^x - o_1^y + \frac{c^{y2} - b^{x2} + c^{x2}}{2 \cdot c^y}, \quad \frac{c^x}{c^y} \cdot o_2^x + o_2^y - \frac{c^y}{2} - \frac{c^{x2}}{2 \cdot c^y}, \\ & - o_1^y + \frac{c^y}{2} + \frac{c^{x2}}{2 \cdot c^y} - \frac{c^x \cdot b^x}{2 \cdot c^y}, \quad o_2^y - \frac{c^y}{2} - \frac{c^{x2}}{2 \cdot c^y} + \frac{c^x \cdot c}{2 \cdot c^y} \} \end{aligned}$$

Коришћењем овог скупа, $o_1^x - o_2^x \overset{*}{\to}_{G'} 0$ се лако може доказати. Заиста,

$$o_1^x - o_2^x \quad \to_{f_1} \quad -o_2^x + \frac{b^x}{2} \quad \to_{f_3} \quad -o_2^x + \frac{b^x}{2} + o_2^x - \frac{b^x}{2} = 0 \, .$$

Слично се може доказати $o_1^y - o_2^y \overset{*}{\to}_{G'} 0$,

$$o_1^y - o_2^y \quad \to_{f_5} \quad -o_2^y + \frac{c^y}{2} + \frac{c^{x2}}{2c^y} - \frac{c^x b^x}{2c^y} \quad \to_{f_6}$$

$$\to_{f_6} \quad -o_2^y + \frac{c^y}{2} + \frac{c^{x2}}{2c^y} - \frac{c^x b^x}{2c^y} + o_2^y - \frac{c^y}{2} - \frac{c^{x2}}{2c^y} - \frac{c^x b^x}{2c^y} = 0 \, .$$

6.3 Формална анализа алгебарских метода у систему *Isabelle/HOL*

Репрезентација планиметријских конструкција коришћењем термова

Прво, било је потребно представити геометријску конструкцију на одговарајући начин тако да се лако може аутоматски обрадити, тј. да се може користити у оквиру нашег алгоритма. Зато су геометријске конструкције репрезентоване термовима. Тренутно, постоје два типа објеката – тачке и праве. Додатно, геометријска тврђења су репрезентована коришћењем термова. У систему *Isabelle/HOL* одговарајући типови се дефинишу на следећи начин:

```
datatype
    point = Point id | Intersect line line | Midpoint point point
and line = Line point point | Bisector point point
       | Normal line point | Parallel line point
and statement = IsIncident point line
       | IsEqualp point point | IsEquall line line
       | IsParallel line line | IsNormal line line
       | IsCongruent point point point point
       | IsBisector line point point
```

Као што можемо видети, тачка може бити дата својим идентификатором или може бити конструисана као пресек две праве или као средиште дужи одређене двема тачкама. Слично, права може бити конструисана као права која је одређена двема тачкама или као симетрала дате дужи итд. Такође, постоје и различита тврђења. На пример, `IsIncident point line` означава тврђење да тачка припада правој, а `IsEqualp point point` означава да су две тачке једнаке.

***Пример* 6.3.1.** *Терм `IsIncident (Point 1) (Line (Point 1) (Point 2))` означава тврђење да тачка припада правој која је одређена том тачком и још једном датом тачком.*

***Пример* 6.3.2.** *Терм*

```
let c = Bisector (Point A) (Point B);
    b = Bisector (Point A) (Point C);
    a = Bisector (Point B) (Point C);
    O1 = Intersect a b;
    O2 = Intersect a c in
    IsEqualp O1 O2
```

је пример који је описан раније – симетрале страница се секу у једној тачки.

Синтетичким термовима који служе за репрезентацију геометријских тврђења може бити дата различита семантика интерпретирањем у различитим моделима геометрије (на пример, Декартова координатна раван, геометрија Хилберта, геометрија Тарског). Коришћењем система модула у систему *Isabelle/HOL* (`locales`) избегава се понављање дефиниција. Зато је дефинисан модуо `AbstractGeometry` који садржи примитивне релације (на пример, релацију инциденције, релацију *између*, релацију *подударно*) и дефинише њихова својства. Изведени концепти се могу дефинисати једино у оквиру овог локала. На пример, изведени појам колинеарност се своди на примитиван појам инциденције — кажемо да су три тачке колинеарне акко постоји права којој припадају све три тачке. Различите геометрије могу интерпретирати овај модуо и (апстрактне) дефиниције изведених појмова се пренесе у те геометрије.

Семантика термова (у апстрактној геометрији) је дата функцијама `point_interp`, `line_interp` и `statement_interp` чији улазни подаци (редом) су `point`, `line` и `statement`, а повратна вредност су редом (апстрактна) тачка, (апстрактна) права или вредност Boolean. Како је апстрактна интерпретација термова јединствено одређена само ако су слободне тачке фиксне, све ове функције имају и додатни аргумент — функцију која пресликава индексе слободних тачака у тачке.

Тврђења се интерпретирају коришћењем примитивних релација апстрактне геометрије, док се конструкције своде на примитивне релације коришћењем Хилбертовог ε оператора (**SOME** у систему *Isabelle/HOL*). На пример:

```
statement_interp (Incident p l) fp =
   incident (point_interp p fp) (line_interp l fp)
point_interp (Intersection l1 l2) fp =
```

```
(SOME P. incident P (line_interp l1 fp)
         ∧ incident P (line_interp l2 fp))
```

Тврђење (записано термом) је исправно у (апстрактној) геометрији ако је тачно за све интерпретације (за било који избор слободних тачака).

```
definition (in AbstractGeometry) valid :: "statement_term => bool"
where
  "valid stmt = (ALL fp. statement_interp stmt fp)"
```

Сви појмови се подижу на ниво конкретног геометријског модела (на пример, Декартова координатна раван, геометрија Хилберта, геометрија Тарског) када се докаже да су интерпретација модула `AbstractGeometry`.

Након што је дефинисана репрезентација геометријских конструкција коришћењем термова, следећи корак је превести термове у скупове полинома тако да се на њих може применити метод Гребнерових база или Вуов метод.

Алгебризација планиметријских термова

Алгоритам се користи да трансформише репрезентацију геометријске конструкције и тврђења из записа коришћењем термова у одговарајуће полиноме. Алгоритам је рекурзиван и његовом применом се добијају два скупа. Први скуп је скуп полинома који репрезентују геометријску конструкцију и зато се зове *скуп–конструкција*. Други скуп је скуп полинома који репрезентују тврђења и њега зовемо *скуп–тврђења*. Метод Гребнерових база се заснива на доказивању да се сваки полином из скупа–тврђења може свести на нула коришћењем Гребнерове базе скупа–конструкција.

Алгоритам рекурзивно обрађује дати терм и за сваки непознати објекат уводи нове координате. Такође, истовремено, додају се нови полиноми у одговарајуће скупове који се заснивају на идентитетима аналитичке геометрије. У сваком тренутку чувају се подаци о тренутном стању, односно о симболичким координатама које су до тог тренутка уведене (што су заправо подтермови полазног терма самог тврђења).

Као пример, доказаћемо кораке алгоритма за тврђење `IsIncident point_t line_t` при чему `point_t` и `line_t` могу бити произвољни, комплексни термови за тачку и праву. Кораци су следећи:

- додају се нове промењљиве x_0 и y_0. Ове промењљиве су непознате координате за тачку O која је дата термом `point_t` — $O(x_0, y_0)$
- додају се промењљиве a_0, b_0, и c_0 које представљају непознате коефицијенте за праву p која је дата термом `line_t` — $p = a_0 \cdot x + b_0 \cdot y + c_0$.
- позива се функција `point_poly(point_t,` x_0`,` y_0`)` која конструише полиноме који спајају промењљиве x_0 и y_0 са термом `point_t`.
- позива се функција `line_poly(line_t,` a_0`,` b_0`)` која конструише полиноме који спајају промењљиве a_0, b_0 и c_0 са термом `line_t`.
- додаје се полином $a_0 \cdot x_0 + b_0 \cdot y_0 + c_0$ у скуп–тврђења.

Као илустрација у наставку се може видети део кода у систему *Isabelle/HOL* који имплементира овај корак превођења.

```
algbrize (IsIncident p l) ==
   "let x = point_id_x 0; y = point_id_y 0;
        a = line_id_a 0; b = line_id_b 0; c = line_id_c 0;
        (s', pp) = point_poly p x y (| maxp = 0, maxl = 0 |);
        (_, lp) = line_poly l a b s' in
    (sup pp lp,
    Fset.Set[poly_of (PSum [PMult[PVar a, PVar x],
                            PMult[PVar b, PVar y]])])"
```

За репрезентацију полинома коришћена је *Isabelle/HOL* теорија *Executable Multivariate Polynomials* [155].

Као што се може приметити постоје две нове функције `point_poly` и `line_poly` које имају два аргумента — термове и две промењљиве. Ове функције су узајамно рекурзивне и користе се да се одреде полиноми скупа–конструкција. Демонстрираћемо како функционишу на следећем примеру — `Intersect line1_t line2_t`. Као и раније, термови `line1_t` и `line2_t` репрезентују линије и могу бити произвољно комплексни. Терм примера репрезентује тачку и потребно је одредити полиноме који одређују ту тачку. Кораци алгоритма у овом примеру су следећи:

- додајемо промењљиве a_1 и b_1 који су коефицијенти праве ($p = a_1 \cdot x + b_1 \cdot y + 1$) која је дата термом `line1_t`

- додајемо промењљиве a_2 и b_2 који су коефицијенти праве ($p = a_2 \cdot x + b_2 \cdot y + 1$) која је дата термом `line2_t`
- позива се функција `line_poly(line_t,` a_1`,` b_1`)`
- позива се функција `line_poly(line_t,` a_2`,` b_2`)`
- додају се полиноми $x \cdot (b_2 \cdot a_1 - a_2 \cdot b_1) - b_1 + b_2$ и $y \cdot (b_2 \cdot a_1 - a_2 \cdot b_1) - (a_2 - a_1)$ у скуп–конструкције.

Ови полиноми су добијени коришћењем геометријских једнакости тако да дато геометријско својство важи.

Описани алгоритам се може оптимизовати и могуће је додати још геометријских објеката (кругови, елипсе итд.) и геометријских тврђења.

Доказивање исправности

Главна идеја је да се аутоматски метод за доказивање теорема у геометрији формално верификује. То значи да је главни део нашег рада да се докаже исправност алгоритма превођења. У намери да се то уради, биће коришћена аналитичка геометрија као веза између синтетичке геометрије и алгебре. Потребно је доказати да све што се докаже коришћењем алгебарских метода такође важи у свим моделима синтетичке геометрије. Са друге стране, још је потребно доказати да је аналитичка геометрија модел синтетичке геометрије и још даље, да су сви модели изоморфни, тј. да све што важи у нашем моделу такође важи и у другим моделима.

Формализацију везе између синтетичке и аналитичке геометрије, односно, доказ да је аналитичка геометрија модел геометрије Тарског и геометрије Хилберта смо показали и дискутовали раније, у поглављу 4.

Веза између аналитичке геометрије и алгебре. Централна теорема коју смо формално доказали је да ако су сви полиноми тврђења нула кад год су и полиноми конструкције нула (тј. ако сви полиноми тврђења припадају идеалу генерисаном над полиномима конструкције), онда је тврђење исправно у аналитичкој геометрији. Односно, ако је неко тврђење доказано методом Гребнерових база оно заиста важи и у аналитичкој геометрији. Формалније записано:

$$(\forall(u,x))(\forall g \in G)((\forall f \in F. f(u,x) = 0) \Rightarrow g(u,x) = 0) \Rightarrow \text{геометријско тврђење}$$

при чему је $F(u,x)$ скуп–конструкције, а $G(u,x)$ је скуп–тврђења. Када кажемо *геометријско тврђење* мислимо на тврђење у аналитичкој геометрији јер су алгебарски методи повезани са аналитичком геометријом. Први део је доказати да важи $(\forall f \in F)(\forall(u,x))f(u,x) = 0$. Други део је доказати да ако је доказано $(\forall g \in G)(\forall f \in F.f(u,x) = 0) \Rightarrow g(u,x) = 0)$ онда геометријско тврђење важи у аналитичкој геометрији.

Запис овог тврђења у систему *Isabelle/HOL* је:

```
theorem "let (cp, sp) = algebrize term in
(ALL ass. ((ALL p : cp. eval_poly ass p = 0) ⟶
           (ALL p : sp. eval_poly ass p = 0)) ⟶
       AnalyticGeometry.valid s)"
```

Доказ се изводи коришћењем индукције у систему *Isabelle/HOL*. Посматрајмо следећи пример:

```
In (Midpoint (Point 0) (Point 1)) (Line (Point 0) (Point 1))
```

`Point 0` и `Point 1` добијају фиксне координате (p_0^x, p_0^y) и (p_1^x, p_1^y). Потом `Midpoint (Point 0) (Point 1)` добија координате (x_1, y_1) и оне су зависне променљиве и зависе од `Point 0` и `Point 1`. На исти начин додељујемо координате (a_1, b_1, c_1) за `Line (Point 0) (Point 1)` (a_1, b_1, c_1) (то су опет зависне променљиве које зависе од већ датих тачака `Point 0` и `Point 1`). За доказ исправности потребно је доказати да важи:

$$2x_1 = p_0^x + p_1^x \qquad a_1(p_1^x p_0^y - p_1^y p_0^x) - c_1(p_1^y - p_0^y) = 0$$
$$2y_1 = p_0^y + p_1^y \qquad b_1(p_1^x p_0^y - p_1^y p_0^x) + c_1(p_1^x - p_0^x) = 0$$

закључак је дат у форми једначине: $\qquad a_1x_1 + b_1y_1 + c_1 = 0$

Ово се лако доказује коришћењем идентитета у аналитичкој геометрији.

Доказивање алгебарског тврђења. На основу претходног доказивање геометријског тврђења своди се на доказивање алгебарског тврђења

$$(\forall(u,x))(\forall g \in G)((\forall f \in F.f(u,x) = 0) \Rightarrow g(u,x) = 0).$$

У пракси се ово ради коришћењем екстерних алгебарских доказивача заснованих на Вуовој методи или на методи Гребнерових база. Иако у систему *Isabelle/HOL* постоји подршка за Гребнерове базе комплекснија тврђења су

ван домашаја те методе. Ако се жели висок степен поузданости онда је неопходно извршити одређену проверу алгебарског доказивача или барем резултата који су од њега добијени. Овај други приступ заснован је на провери сертификата и описан је у радовима [67, 129]. Наиме, приликом доказивања алгебарског тврђења, у виду матрице памте се трансформације које је направио алгебарски доказивач. Потом се у оквиру асистента за доказивање теорема *Coq* проверава да ли се применом те матрице трансформација на дате полазне полиноме заиста добија нула полином. Ми се у оквиру ове тезе нисмо бавили оваквим аспектима повезивања доказивача и асистента за доказивање теорема.

Веза између синтетичке геометрије и алгебре. Уколико покажемо да тврђење важи у Декартовој координатној равни, односно у аналитичкој геометрији, да ли из тога следи да се тврђење може доказати из аксиома Хилберта или из аксиома Тарског? Постоји тврђење да су сви модели аксиома Хилберта међусобно изоморфни. Слично тврђење постоји и за моделе аксиома Тарског. Поред тога, показано је и да су аксиоматски системи Тарског и Хилберта потпуно дедуктивни, што значи да свако тачно тврђење у моделу може да се докаже из аксиома. На основу тога знамо да ако алгебарски доказивач покаже одговарајућу везу између полинома, онда то тврђење важи у Декартовој равни (аналитичкој геометрији), па самим тим у свим осталим моделима геометрије, што повлачи да се то тврђење може доказати из, на пример, Хилбертових аксиома. Формализација ових мета–теоретских особина геометрије је веома захтеван подухват и било би потребно формализовати појам доказивости у оквиру геометрије Хилберта или у оквиру геометрије Тарског. Ипак, ова теза прави одређене кораке у том смеру, а то је формалан доказ да Декартова раван представља модел геометрије Хилберта и модел геометрије Тарског.

6.4 Примена алгебарских метода на проблеме у стереометрији

Коришћење алгебарских доказивача се интензивно користи у планиметрији и постоји много система посвећених овом проблему. То је разлог зашто смо у претходном поглављу разматрали теоријске аспекте алгебарских доказива-

ча. Са друге стране, коришћење алгебарских доказивача у стереометрији није значајно истраживано и према нашем сазнању, не постоји потпуно развијен аутоматски систем за коришћење алгебарских доказивача за доказивање у стереометрији. Стога нам је циљ у овом поглављу да дизајнирамо систем за запис и трансформацију геометријских тврђења на начин погодан за примену у оквиру алгебарских доказивача. Теоријска анализа таквог стереометријског система, слична оној каква је урађена за планиметријске доказиваче, остављена је за даљи рад.

Аутоматско доказивање у стереометрији – досадашњи резултати. Чу и сарадници су представили метод запремине за решавање проблема у стереометрији [31]. То је полу–алгебарски метод који је проширење методе површине за стереометрију. Хипотезе се могу конструктивно представити, а закључци су полиномијалне једначине које садрже неколико геометријских величина, као што су однос запремина, однос дужи, однос површина и Питагорине разлике. Кључна идеја метода је да елиминише тачке из закључка геометријског тврђења коришћењем неколико основних својстава запремине.

Главна мотивација за наш рад потекла је из интересантног рада који представља неколико примера алгебарског доказивања у стереометрији [152]. У раду се посматрају задаци из стереометрије са Олимпијских такмичења из математике. Они у раду представљају три различита проблема и дају полиноме које су извели на папиру и који описују посматране проблеме. Коришћењем ова три примера они показују да се алгебарски методи могу користити за доказивање у стереометрији. За сваки пример користили су три различита метода: метод карактеристичног скупа [171, 162, 50, 26], метод Гребнерових база [41, 98, 156, 35] и метод вектора [106]. Методи се пореде и закључак је да метод вектора даје бољи геометријски доказ, али формуле могу бити дуге и незгодне за манипулацију и израчунавање. Ипак, они не нуде неки систематичан начин како се геометријска тврђења могу представити полиномима.

Према нашем досадашњем знању, не постоје радови који описују начин примене Вуове методе или методе Гребнерових база на проблеме из стереометрије.

Динамички геометријски софтвер. Последњих неколико година, рачунари и технологија се интензивно користе и мењају начин како се предаје

геометрија. Динамички геометријски системи као што су *GeoGebra* [2], *Cinderella* [3], *Geometer's Sketchpad* [4], *Cabri* [5], *Eukleides* [6] се данас често користе у свим нивоима образовања. Студенти користе овакве системе да би изводили геометријске конструкције и дијаграме које могу да мењају променом слободних тачака. Такви динамички дијаграми су бољи него статичке слике јер померање слободних тачака може да пружи додатни увид у проблем и да открије дегенерисане случајеве и да помогне студентима да утврде да ли је нешто тачно ако и само ако је неки специјални размештај тачака задат (на пример, неко својство може бити тачно само ако је нека тачка између неке друге две тачке, а нетачно је ако то није случај, неко својство може бити тачно само за оштре, али не и за тупе углове, итд.).

Интензивним мењањем дијаграма померајући слободне тачке, студент може бити прилично сигуран да ли је својство тачно у општем случају (тј. тачно у готово свим случајевима, осим у малој групи дегенерисаних случајева), али ипак, то не можемо сматрати доказом и овакав приступ је подложан грешкама. Зато, у скорије време, динамички геометријски системи су проширени аутоматским системима за доказивање, који аутоматски могу доказати тврђење о конструисаним објектима [16]. Такви системи су најчешће алгебарски (операције се изводе над симболичким координатама геометријских објеката).

Можда још значајна употреба динамичког геометријског софтвера може бити за тродимензионални простор у коме је често тешко голим оком одредити неко својство. Ово је најчешће стога што се тродимензиони простор посматра као дводимензиона пројекција, па самим тим мере и односи нису у складу са стварним дијаграмом. Неки системи су почели да развијају подршку за тродимензионе конструкције. У најновијој верзији система *GeoGebra* развијена је подршка за динамичку тродимензионалну геометрију и графику[7]. Могуће је креирати и интерактивно мењати тродимензионалне објекте као што су тачке, праве, полигони, сфере, као и тродимензионе цртеже функција. Ипак, овај систем не подржава доказивање тврђења о тродимензионалним објектима.

Такође, постоји и додатак за систем *Cinderella*, *Cindy3D* [8]. Могуће је цртати

[2] https://www.geogebra.org/
[3] https://www.cinderella.de/tiki-index.php
[4] http://www.dynamicgeometry.com/
[5] http://www.cabri.com/
[6] http://www.eukleides.org/
[7] https://wiki.geogebra.org/en/3D_Graphics_View
[8] http://gagern.github.io/Cindy3D/

објекте коришћењем команди и формула које их описују.

Већина истраживања како у динамичким геометријским системима, као и у аутоматским доказивачима теорема је посвећена само дводимензионалној Еуглидској геометрији (планарној геометрији). Иако постоји неколико покушаја да се примене алгебарски доказивачи теорема на тродимензионалну еуклидску просторну геометрију (стереометрију), ми нисмо нашли да постоји детаљни опис ових метода, нити јавно доступних аутоматских доказивача за стереометрију. У овом раду ми истражујемо и поредимо неколико приступа како се алгебарски доказивачи засновани на Вуовој методи и методи Гребнерових база могу применити на проблеме из стереометрије. Нудимо један систем за стереометрију који може да доказује тврђења о својствима конструисаних објеката. Такође, анализирамо корпус проблема из стереометрије и оцењујемо коришћене методе. Дискутујемо о изазовима и могућим применама у пољу предавања геометрије.

Алгебризација геометријских релација у стереометрији

Да бисмо могли да применимо алгебарске методе прво се мора омогућити репрезентација различитих геометријских релација између објеката у стереометрији коришћењем полиномијалних једнакости над њиховим координатама. У овом поглављу даћемо примере како се то може учинити за најчешће релације међу објектима.

Постоји више приступа за запис релација као полиномијалних једнакости и прво питање које се поставља је који објекти у тродимензионом простору се сматрају основним. У првом приступу за који смо се одлучили, сви објекти су дефинисани коришћењем тачака (на пример, праве се дефинишу преко две различите тачке, равни се дефинишу преко три различите, неколинеарне тачке итд.). Једине промењљиве које се користе у свим полиномима су координате тачака. У другом приступу, све врсте објекта се представљају коришћењем њихових сопствених координата (на пример, права се дефинише координатама једне своје тачке и координатама вектора правца, а раван се дефинише помоћу коефицијената једначине равни – координате вектора нормале на раван и њена удаљеност од координатног почетка). Полиноми укључују све ове координатне промењљиве. Показаћемо како енкодирати релације коришћењем оба приступа и упоредићемо њихову ефикасност.

Основни појмови коришћени у конструкцији полинома

Већина релација се изражава коришћењем истог скупа појмова које ћемо овде увести.

Сваки објекат је представњен неком n–торком параметара (а видећемо да то могу бити и симболичке и нумеричке вредности).

Тачке имају три параметра, означена са $([\_]^x, [\_]^y, [\_]^z)$ који репрезентују њене координате. Свака тачка је задата или својим симболичким или нумеричким координатама. За сваку новоуведену тачку се додељују нове симболичке координате.

Праве се представљају различито у зависности од коришћеног приступа. У првом приступу, права је задата са две различите тачке и са шесторком која представља координате тих тачака. За праву p, прва тачка ће бити означена са p_A, а друга тачка ће бити означена са p_B.

У другом приступу, права је дата датом тачком A и датим вектором v. Вектор праве p ће бити означен са $\overrightarrow{p_v}$, а тачка праве p ће бити означена са p_A. Зато, као и у првом приступу, права ће и у другом приступу имати шест параметара који су означени са $([\_]^{v_x}, [\_]^{v_y}, [\_]^{v_z}, [\_]^{A_x}, [\_]^{A_y}, [\_]^{A_z})$, који репрезентују праву дату једначином:

$$x = k \cdot [\_]^{v_x} + [\_]^{A_x} \quad y = k \cdot [\_]^{A_y} + [\_]^{A_y} \quad z = k \cdot [\_]^{v_z} + [\_]^{A_z}.$$

У претходној једначини, k означава размеру праве, али ова информација се не чува међу параметрима праве (а то је поменута шесторка). У неким полиномима ће бити потребно користити параметар размере праве, али ће се посматрати као нова симболичка променљива.

Равни се такође могу различито представити у зависности од коришћеног приступа. У првом приступу, раван се задаје са три неколинеарне тачке, односно са деветорком њихових координата. Прва тачка равни π ће бити означена са π_A, друга тачка ће бити означена са π_B, а трећа тачка ће бити означена са π_C.

У другом приступу, равни су одређене нормалним вектором v и додатним параметром d (померај у односу на координатни почетак). Вектор равни π ће бити означен са $\overrightarrow{\pi_v}$, а слободан параметар равни ће бити означен са $[\_]^d$. Стога ће раван имати само четири параметра, означена са $([\_]^{v_x}, [\_]^{v_y}, [\_]^{v_z}, [\_]^d)$, који репрезентују раван дату следећом једначином:

$$[\_]^{v_x} \cdot x + [\_]^{v_y} \cdot y + [\_]^{v_z} \cdot z + [\_]^d = 0.$$

Вектор одређен двема тачкама $A = (a^x, a^y, a^z)$ и $B = (b^x, b^y, b^z)$ је $\overrightarrow{AB} = (b^x - a^x, b^y - a^y, b^z - a^z)$. Стандардни појмови скаларног производа, векторског производа и мешовитог производа се могу применити над векторима. Скаларни производ вектора $v = (v^x, v^y, v^z)$ и $u = (u^x, u^y, u^z)$ је $v \cdot u = v^x \cdot u^x + v^y \cdot u^y + v^z \cdot u^z$, њихов векторски производ је одређен матрицом:

$$v \times u = \begin{vmatrix} \overrightarrow{i} & \overrightarrow{j} & \overrightarrow{k} \\ v^x & v^y & v^z \\ u^x & u^y & u^z \end{vmatrix},$$

а мешовити производ са вектором $w = (w^x, w^y, w^z)$ је једнак $v \cdot (u \times w)$, и одређен је матрицом:

$$\begin{vmatrix} v^x & v^y & v^z \\ u^x & u^y & u^z \\ w^x & w^y & w^z \end{vmatrix}.$$

Репрезентација стереометријских релација

У овом поглављу су дати полиноми који аритметички описују релације над конструисаним објектима (на пример, две тачке су једнаке, две линије су паралелне, две равни су нормалне). Свака релација уводи полиномијална ограничења над координатама објеката који учествују у релацији и у зависности од приступа могу бити различити.

Улазни параметри дате релације су параметри свих објеката који су укључени у ту релацију. На пример, за релацију `congruent` $A\ B\ C\ D$ улаз су четири тачке, A, B, C, and D, односно њихове симболичке координате: (a^x, a^y, a^z), (b^x, b^y, b^z), (c^x, c^y, c^z) и (d^x, d^y, d^z).

▷ `congruent` $A\ B\ C\ D$

Опис: Дужи AB и CD су подударне.

Полиноми:
$\overrightarrow{AB} \cdot \overrightarrow{AB} = \overrightarrow{CD} \cdot \overrightarrow{CD}$.
Приметимо да из претходног израза можемо добити полиномијалну једнакост $poly = 0$, при чему је

$$poly = (a^x - b^x)^2 + (a^y - b^y)^2 + (a^z - b^z)^2 - (c^x - d^x)^2 - (c^y - d^y)^2 - (c^z - d^z)^2$$

Објашњење: Квадрати растојања између A и B мора бити једнако квадрату растојања између C и D.

▷ `segments_in_ratio` $A\ B\ C\ D\ m\ n$

Опис: Дужина дужи AB и CD су у датом односу $\frac{m}{n}$, тј. $\frac{|AB|}{|CD|} = \frac{m}{n}$.

Полиноми:
$n^2 \cdot \overrightarrow{AB} \cdot \overrightarrow{AB} = m^2 \cdot \overrightarrow{CD} \cdot \overrightarrow{CD}$.

Објашњење: Квадрати растојања између A и B и између C и D морају бити у односу $\frac{m^2}{n^2}$. Приметимо да се ово своди на подударност када је $m = n$.

▷ `is_midpoint` $M\ A\ B$

Опис: Проверава да ли је тачка M средња тачка дужи одређене тачкама A и B.

Полиноми: $\overrightarrow{MA} = \overrightarrow{MB}$. Приметимо да ова једнакост даје три различита полинома $poly_1 = 0$, $poly_2 = 0$, $poly_3 = 0$:

$poly_1 = 2m^x - a^x - b^x$
$poly_2 = 2m^y - a^y - b^y$
$poly_3 = 2m^z - a^z - b^z$

▷ `point_segment_ratio` $M\ A\ B\ p\ q$

Опис: Проверава да ли тачка M дели дуж одређену тачкама A и B у односу који је одређен са p и q, тј. $\frac{|MA|}{|MB|} = \frac{p}{q}$.

Полиноми: Изводе се три полинома из: $q \cdot \overrightarrow{MA} = p \cdot \overrightarrow{MB}$.

Објашњење: Приметимо да `is_midpoint` $M\ A\ B$ може такође бити записано коришћењем овог правила, на следећи начин `point_segment_ratio` $M\ A\ B\ 1\ 1$.

▷ `equal_points` $A\ B$

Опис: Проверава да ли две тачке A и B имају исте координате.

Полиноми: Изводе се три полинома из израза $\overrightarrow{AB} = 0$.

▷ `translate` $A\ O\ v$

Опис: Проверава да ли је тачка O једнака тачки која се добија транслирањем тачке point A за вектор (v^x, v^y, v^z).

Полиноми: Полиноми су изведени из $\overrightarrow{AO} = (v^x, v^y, v^z)$

▷ `orthogonal_4points` $A\ B\ C\ D$

Опис: Проверава да ли је права одређена са тачкама A и B нормална на праву одређену са тачкама C и D.

Полиноми: $\overrightarrow{AB} \cdot \overrightarrow{CD} = 0$

▷ `orthogonal_lines` $p\ q$

Опис: Две праве, p и q су нормалне.

Полиноми: Ако је коришћен први приступ, онда су праве задате тачкама (p_A, p_B) и (q_A, q_B), па се то своди на претходни случај и полином је $\overrightarrow{p_A p_B} \cdot \overrightarrow{q_A q_B} = 0$. Ако се користи други приступ, као улаз задат је вектор правца правих и полином је $\overrightarrow{p_v} \cdot \overrightarrow{q_v} = 0$.

▷ `incident` $A\ p$

Опис: Проверава да ли тачка A припада правој p.

Полиноми: Ако је коришћен први приступ, онда се изводе три полинома из једнакости $\overrightarrow{p_A p_B} \times \overrightarrow{Ap_A} = 0$. Ако је коришћен други приступ, онда се изводе три полинома из једнакости $\overrightarrow{p_v} \times \overrightarrow{Ap_A} = 0$.

▷ `parallel_lines` $p\ q$

Опис: Проверава да ли су две праве, p и q паралелне.

Полиноми: Изводе се три полинома из $\overrightarrow{p_A p_B} \times \overrightarrow{q_A q_B}$ или из $\overrightarrow{p_v} \times \overrightarrow{q_v}$ у зависности од приступа.

▷ `line_orth_plane_4points` $l\ A\ B\ C\ D$

Опис: Проверава да ли је права l која садржи тачку D нормална на раван одређеном тачкама A, B и C.

Полиноми: Изводе се три полинома из $\overrightarrow{l_v} = \overrightarrow{AB} \times \overrightarrow{AC}$ и три полинома из једнакости $\overrightarrow{l_v} \times \overrightarrow{Dl_A} = 0$..

▷ `parallel_planes` α β

Опис: Проверава да ли су две равни α и β паралелне.

Полиноми: Ако се користи други приступ, изводе се три полинома из $\overrightarrow{\alpha_v} \times \overrightarrow{\beta_v} = 0$.

Ако се користи први приступ изводе се полиноми из једнакости:

$$\overrightarrow{\beta_A\beta_B} \cdot \overrightarrow{\alpha_A\alpha_C} \times \overrightarrow{\alpha_B\alpha_A} = 0$$

$$\overrightarrow{\beta_A\beta_C} \cdot \overrightarrow{\alpha_A\alpha_C} \times \overrightarrow{\alpha_A\alpha_B} = 0$$

▷ `orthogonal_planes` α β

Опис: Проверава да ли су две равни, α и β нормалне.

Полиноми: Ако се користи први приступ полином се изводи из: $(\overrightarrow{\alpha_A\alpha_B} \times \overrightarrow{\alpha_A\alpha_B}) \cdot (\overrightarrow{\beta_A\beta_B} \times \overrightarrow{\beta_A\beta_B}) = 0$.

Ако се користи други приступ, полином се изводи из: $\overrightarrow{\alpha_v} \cdot \overrightarrow{\beta_v} = 0$.

▷ `point_in_plane` A π

Опис: Проверава да ли тачка A припада равни π.

Полиноми: Ако је коришћен први приступ, онда се полином добија из:

$$\overrightarrow{\pi_A A} \cdot (\overrightarrow{\pi_A\pi_B} \times \overrightarrow{\pi_A\pi_C}) = 0.$$

Ако је коришћен други приступ, полином се добија из једнакости: $\overrightarrow{\pi_v} \cdot \overrightarrow{A} + \pi^d = 0$.

▷ `parallel_line_plane` p α

Опис: Проверава да ли су права p и раван α паралелни.

Полиноми: Ако је коришћен први приступ, једнакост је: $\overrightarrow{p_Ap_B} \cdot (\overrightarrow{\alpha_A\alpha_B \times \alpha_A\alpha_C}) = 0$.

Ако је коришћен други приступ, полином се добија из једнакости: $\overrightarrow{p_v} \cdot \overrightarrow{\alpha_v} = 0$.

Примедба: Ова релација важи и у случају када права припада равни.

▷ `orthogonal_line_plane` p α

Опис: Проверава да ли су права p и раван α нормални.

Полиноми: Ако је коришћен први приступ, полиноми се изводе из две једнакости: $\overrightarrow{p_Ap_B} \cdot \overrightarrow{\alpha_A\alpha_B} = 0$ and $\overrightarrow{p_Ap_B} \cdot \overrightarrow{\alpha_A\alpha_C} = 0$.

Ако је коришћен други приступ, три полинома се изводе из једнакости: $\overrightarrow{p_v} \times \overrightarrow{\alpha_v} = 0$.

▷ `equal_angles` $A\ O\ B\ C\ K\ D$

Опис: Проверава да ли су два угла $\angle AOB$ и $\angle CKD$ једнаки.

Полиноми: Ова релација се може изразити коришћењем тригонометрије. Ипак, треба имати на уму да коришћењем тригонометрије, услов је ослабљен јер се пореде косинуси углова, а као што је познато, косинус тупог угла и косинус оштрог угла могу бити исти, а углови (јасно) нису једнаки. Полиноми се изводе из

$$\cos^2 \angle AOB = \cos^2 \angle CKD.$$

А косинус угла се може одредити на следећи начин:

$$\cos^2 \angle AOB = \frac{(\overrightarrow{AO} \cdot \overrightarrow{BO})^2}{|AO|^2|BO|^2}$$

при чему је $|AO|^2 = \overrightarrow{AO} \cdot \overrightarrow{AO}$. Једнакост за $\cos^2 \angle CKD$ је слична. Коначно, након неколико једноставних алгебарских операција, полином релације се може извести из једнакости

$$(\overrightarrow{AO} \cdot \overrightarrow{BO})^2|CK|^2|DK|^2 = (\overrightarrow{CK} \cdot \overrightarrow{DK})^2|AO|^2|BO|^2.$$

Ипак, као што ћемо видети у наредном поглављу, овако задат полином је веома комплексан и било је потребно раставити га на једноставније да би могао ефикасно да се користи у доказивачима теорема.

Тела. Тела се задају коришћењем релација које важе за њихова темена и за њихове странице. Одлучили смо да подржимо само она тела која се налазе у неком *канонском* положају — на пример, при дефинисању коцке, једно теме се налази у координатном почетку, а друга три темена се налазе на координатним осама (x–оси, y–оси и z–оси). Ипак, овакав приступ има мане. На пример, није могуће задати више од једне коцке коришћењем елементарне наредбе за задавање коцке. Темена других

коцки на слици које нису у канонском положају морају се задати коришћењем релација које смо представили изнад. Ипак, са друге стране, у геометријским проблемима која се сусрећу у збиркама најчешће постоји само једно слободно тело, и без губитка на општости се може претпоставити да је оно у канонском положају. Када се у тексту задатка уводе друга тела, она су обично зависна у односу на већ задато слободно тело, па су њихова темена у некој релацији са већ датим објектима. Зато, могућност задавања само канонских објеката за већину задатака није представљао проблем.

▷ `make_cube` $A\ B\ C\ D\ A_1\ B_1\ C_1\ D_1$

Опис: Коцка у канонском положају, са дужином странице једнакој 1.

Објекти и параметри: Тачке $A(0,0,0)$, $B(1,0,0)$, $C(1,1,0)$, $D(0,1,0)$, $A_1(0,0,1)$, $B_1(1,0,1)$, $C_1(1,1,1)$ и $D(0,1,1)$.

Полиноми: Не генеришу се полиноми.

Објашњење: Како је коцка у канонском положају, не уводе се нове симболичке променљиве.

▷ `make_tetrahedron` $A\ B\ C\ D$

Опис: Тетраедар у канонском положају.

Објекти и параметри: Темена тетраедра имају координате $A(0,0,0)$, $B(1,0,0)$, $C(c^x, c^y, 0)$ и $D(c^x, d^y, d^z)$, при чему се уводе четири нова параметра.

Полиноми:
$$poly_1 = 2 \cdot c^x - 1$$
$$poly_2 = 2 \cdot c^{y2} - 3$$
$$poly_3 = 3 \cdot d^y - c^y$$
$$poly_4 = 3 \cdot d^{z2} - 2$$

Објашњење: $c^x = \frac{1}{2}$, $c^y = \frac{\sqrt{3}}{2}$, $d^y = \frac{\sqrt{3}}{6} = \frac{c^y}{3}$, $d^z = \frac{\sqrt{2}}{\sqrt{3}}$. Приметимо да сви објекти имају или симболичке или бројевне параметре. Коефицијенти полинома морају увек бити цели бројеви, па се ирационалне вредности (као и разломци) морају увести коришћењем полинома.

▷ `make_pyramid_4side` $A\ B\ C\ D\ S$

Опис: Правилна пирамида у канонском положају – основа пирамиде је јединични квадрат у xOy равни, висина пирамиде није фиксирана, а стране пирамиде се једнаке дужине.

Објекти и параметри: Тачке $A(0,0,0)$, $B(1,0,0)$, $C(1,1,0)$, $D(0,1,0)$ и $S(s^x, s^y, s^z)$, са три нове симболичке променљиве s^x, s^y и s^z.

Полиноми: $poly_1 = 2 \cdot s^x - 1$
$poly_2 = 2 \cdot s^y - 1$

Објашњење: Пројекција врха пирамиде је $(s^x, s^y, 0)$ и она лежи у центру јединичног квадрата, тако да $s^x = s^y = \frac{1}{2}$. Приметимо да s^z није ограничено.

▷ `make_square` A B C D

Опис: Задаје се квадрат у канонском положају, односно квадрат у равни xOy, чије једно теме је у координатном почетку, а друга два темена на координатним осама и дужина странице је једнака 1.

Објекти и параметри: Тачке $A(0,0,0)$, $B(1,0,0)$, $C(1,1,0)$, $D(0,1,0)$.

Полиноми: Не креирају се нови полиноми.

▷ `equilateral_triangle` A B C

Опис: Једнакостранични троугао у канонском положају – налази се у xOy равни, једна тачка је у координатном почетку, а друга тачка је на x–оси.

Објекти и параметри: Тачке $A(0,0,0)$, $B(1,0,0)$, $C(c^x, c^y, 0)$, са два нова параметра c^x и c^y.

Полиноми: $poly_1 = 2 \cdot c^x - 1$
$poly_2 = 4 \cdot c^y - 3$

▷ `regular_hexagon` A_1 A_2 A_3 A_4 A_5 A_6

Опис: Правилни шестоугаоник у канонском положају – налази се у xOy равни, једна тачка је у координатном почетку, а друга тачка је на x–оси.

Објекти и параметри: Тачке $A_1(0,0,0)$, $A_2(1,0,0)$, $A_3(a_3^x, a_3^y, 0)$, $A_4(1, a_4^y, 0)$, $A_5(0, a_4^y, 0)$ и $A_6(a_6^x, a_3^y, 0)$, са четири нова параметра a_3^x, a_3^y, a_4^y и a_6^x.

$$\begin{aligned} \textit{Полиноми: } poly_1 &= 2 \cdot a_3^x - 3 \\ poly_2 &= 4(a_3^y)^2 - 3 \\ poly_3 &= a_4^y - 3 \\ poly_4 &= 2a_6^x - 1 \end{aligned}$$

Упрошћавање полинома

Да бисмо могли да тестирамо предложену алгебризацију геометријских релација користили смо две алатке, *GeoProver* и *Mathematica*, метод Гребнерових база. Први проблем на који смо наишли је временско и просторно ограничење које се дешавало у бројним ситуацијама због комплексности скупа полинома. Разлог за ово је велики број промењљивих и велики број полинома који су у процесу алгебризације креирани. Треба имати на уму да су полиноми у стереометрији доста комплекснији од одговарајућих полинома у планарној геометрији. Зато смо морали да упростимо скуп добијених полинома.

Користили смо приступ о којем смо раније доста писали – Харисонов приступ без губитка на општости [78] тако што смо бирали погодни координатни систем јер избор координатног система може значајно да смањи комплексност полинома.

За три независно задате тачке, A, B и C, могуће је изабрати њихове координате на такав начин да је $A(0,0,0)$ у координатном почетку, $B(0,0,b^z)$ се налази на z–оси, а $C(0,c^y,c^z)$ лежи у yOz равни. Са овим избором координата, број промењљивих је смањен за шест, а одговарајуће нуле значајно поједностављују полиноме. Овај приступ је често коришћен у алгебарским методима. Не утиче на општост тврђења и оправданост коришћења овог приступа је у чињеници да транслације и ротације могу бити коришћење да трансформишу тачке у њихове канонске положаје. Транслације и ротације су изометрије што значи да чувају растојање, али и бројне геометријске релације као што је инциденција, нормалност, паралелност, меру угла и друге.

Без примене овог метода, чак и најједноставнија тврђења не могу бити доказана. Избором погодних координата значајно се повећава једноставност система полинома и тиме алгебарски доказивачи су знатно ефикаснији. Даље, може се десити да су неки полиноми вишак јер постану једнаки 0 и онда је само потребно избрисати их из система. Додатно, неки полиноми постану полиноми који имају само једну промењљиву на неки степен (имају само један

моном) и ти полиноми исто могу бити избрисани, а одговарајућа променљива се може поставити на нула.

Поред поменутог, треба још имати на уму да доказивач *GeoProver* не може да трансформише систем полинома који садржи једнакост $0 = 0$ у троугаони систем. Са друге страна, метод Гребнерових база имплементиран у систему *Mathematica* није имао проблема приликом доказивања уколико су се у систему налазили полиноми $0 = 0$ или $c_i \cdot x_i^s = 0$.

Раније смо већ видели како се погодно могу задати тела. О томе смо писали у одељку 6.4 и управо избор да тела ставимо у канонски положај значајно утиче на упрошћавање полинома.

Надаље, применили смо још један метод за симплификацију и то у случају када имамо полиноме облика $c_i \cdot x_i - c_j \cdot x_j = 0$ или $c_i \cdot x_i + c_j \cdot x_j = 0$. Ако без губитка на општости претпоставимо да $j < i$, заменимо свако појављивање x_i са x_j у првом случају, односно свако појављивање x_i са $-x_j$, онда можемо и поменуте полиноме избрисати из система. Овим се смањује и број полинома и број променљивих.

Један од најкомплекснијих полинома који се могу добити током алгебризације је полином који се добија од релације „једнаки углови". Приликом доказивања тврђења о једнакости углова, доказивач *GeoProver* је достигао просторни лимит већ након неколико корака. Са друге стране, метод Гребенерових база из система *Mathematica* је радио неколико сати након чега смо решили да прекинемо доказивање (а нисмо добили одговор да ли је тврђење тачно). Потом је полином за једнакост углова подељен у више мањих, једноставнијих полинома:

$$
\begin{aligned}
scalar_1 &= \overrightarrow{AO} \cdot \overrightarrow{BO} \\
scalar_2 &= \overrightarrow{CK} \cdot \overrightarrow{DK} \\
dist_{CK} &= |CK| \\
dist_{DK} &= |DK| \\
dist_{AO} &= |AO| \\
dist_{BO} &= |BO| \\
poly = \quad & scalar_1 * scalar_1 * dist_{CK} * dist_{DK} \\
& -scalar_2 * scalar_2 * dist_{AO} * dist_{BO}
\end{aligned}
$$

Иако смо овим повећали скуп полинома, оба доказивача нису имала проблема приликом доказивања истог тврђења и оба су доказивање завршили у

кратком временском периоду, око једне секунде.

Додатно подешавање полинома за Вуов метод

Најинтересантнији пример приликом задавања релација које задовољава тачка је случај пресека правих. У тродимезионом простору немају све праве пресек, неке су паралелне, али неке могу бити и мимоилазне. Погледајмо релацију и полиноме које она генерише.

`intersection_lines` $A\ l_1\ l_2$

Опис: Тачка A је пресек две дате праве l_1 и l_2.

Улаз: Две дате праве l_1 и l_2 задате својим параметрима (посматрамо само други приступ, слична ситуација је и у првом приступу) $l_1(l_1^{v_x}, l_1^{v_y}, l_1^{v_z}, l_1^{p_x}, l_1^{p_y}, l_1^{p_z})$ и $l_2(l_2^{v_x}, l_2^{v_y}, l_2^{v_z}, l_2^{p_x} l_2^{p_y}, l_2^{p_z})$.

Нови објекти и параметри: Тачка A са симболичким координатама (a^x, a^y, a^z). Параметри k_1 и k_2 који редом представљају размере правих l_1 и l_2.

Полиноми: Полиноми се генеришу коришћењем правила: `intersection_lines` $A\ l_1\ l_2$ = `incident` $A\ l_1$ и `incident` $A\ l_2$

Полиноми:
$$
\begin{aligned}
poly_1 &= a^x - k_1 \cdot l_1^{v_x} - l_1^{p_x} \\
poly_2 &= a^y - k_1 \cdot l_1^{v_y} - l_1^{p_y} \\
poly_3 &= a^z - k_1 \cdot l_1^{v_z} - l_1^{p_z} \\
poly_4 &= a^x - k_2 \cdot l_2^{v_x} - l_2^{p_x} \\
poly_5 &= a^y - k_2 \cdot l_2^{v_y} - l_2^{p_y} \\
poly_6 &= a^z - k_2 \cdot l_2^{v_z} - l_2^{p_z}
\end{aligned}
$$

Објашњење: Како тачка A припада обема правама l_1 (која је дата тачком point $\overrightarrow{l_1^p}$ и вектором $\overrightarrow{l_1^v}$) и l_2 (дата тачком $\overrightarrow{l_2^p}$ и вектором $\overrightarrow{l_2^v}$), онда тачка A мора да задовољи њихове параметарске једначине, односно, мора да важи $\overrightarrow{A} = \overrightarrow{l_1^p} + k_1 \cdot \overrightarrow{l_1^v}$ и $\overrightarrow{A} = \overrightarrow{l_2^p} + k_2 \cdot \overrightarrow{l_2^v}$.

Оно што је важно је да ови полиноми користе две нове променљиве k_1 и k_2 које редом представљају размере правих l_1 и l_2. То значи да је укупан број нових променљивих пет, а постоји шест полинома које ова релација генерише. Ако су свих шест полинома укључени у скуп полинома, онда није могуће одредити еквивалентни троугаони систем јер има више полинома него непознатих променљивих. Довољно је пет полинома да би се одредило решење система, односно вредности свих параметара, а преостали, шести полином је

у функцији оправдавања решења. Односно, уколико су тачке мимоилазне, за израчунате вредности шести полином неће бити једнак нули, иако су остали полиноми једнаки нули. Било који од задатих шест полинома може бити искључен из система, али мора бити проверен коришћењем доказивача да ли је и он нула (ако није, онда тврђење у општем случају не важи).

Ипак, у зависности од система, треба бити пажљив приликом избора који полином избацити из скупа полинома. На пример, нека је $(1, 1, 0)$ вектор праве l_1, а тачка која јој припада је $(0, 0, 0)$ и нека је $(0, 0, 1)$ вектор праве l_2, а права садржи тачку $(1, 1, 1)$. Тада, систем добијених полинома је:

$$\begin{aligned} \textit{Polynomials: } & poly_1 = a^x - k_1 \\ & poly_2 = a^y - k_1 \\ & poly_3 = a^z \\ & poly_4 = a^x - 1 \\ & poly_5 = a^y - 1 \\ & poly_6 = a^z - k_2 - 1 \end{aligned}$$

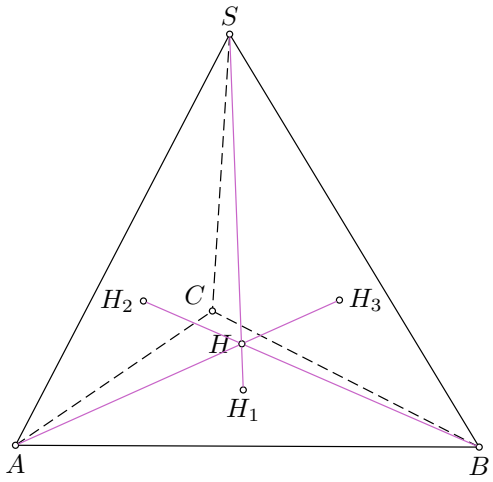
Као што се може приметити, једини полином који има променљиву k_2 је последњи полином. Зато, последњи полином се не сме избацити из скупа полинома, а могуће решење је избацити полином $poly_5$. Иако се све сличне специфичне ситуације могу лако детектовати, имплементација избора који полином избацити, а које полиноме задржати је прилично незгодна јер је потребно много *if–else* испитивања.

Сличан проблем је и код релације `intersection_4points` $M\ A\ B\ C\ D$ и на истоветан начин се и овај проблем решава.

Експерименти

У овом поглављу ћемо представити резултате тестирања алгебризације коришћењем система *GeoProver* и Гребнерових база имплементираних у систему *Mathematica*. Применили смо методе на двадесет и пет проблема из збирке задатака „Збирка задатака из геометрије простора за припрему пријемног испита на Архитектонском факултету" [174], на три задатка из „Збирке задатака из геометрије" [85] и на проблем једнакости углова представљен у раду о примени аутоматских доказивача на проблеме са Олимпијада из математике [152].

Пример нормала тетраедра. *Нека је $ABCS$ тетраедар и нека су h_a, h_b, h_c и h_s редом висине из темена тетраедра A, B, C и S на одговарајуће наспрамне странице и нека је H пресек h_a и h_b. Тада H припада и висинама h_s и h_c.*



Слика 6.3: Нормале тетраедра се секу у истој тачки

Тврђење се може записати на следећи начин:

```
make_tetrahedron A B C S

plane_points α1 A B C
plane_points α2 B C S
plane_points α3 A C S
plane_points α4 A B S

orthogonal_line_plane h1 α1 S
orthogonal_line_plane h2 α2 A
orthogonal_line_plane h3 α3 B
orthogonal_line_plane h4 α4 C

intersection_lines H1 h1 h2
intersection_lines H2 h2 h3
intersection_lines H3 h2 h4

equal_points H1 H2
equal_points H1 H3
```

Како је тетраедар у канонском положају према самој конструкцији, координате тачака су $A(0,0,0)$, $B(0,0,1)$, $C(x_1,x_2,0)$ и $S(x_3,x_4,x_5)$ при чему су x_1, x_2, x_3, x_4 и x_5 зависне променљиве и налазе се у полиномима који описују тетраедар (погледати у ранијем поглављу полиноме за тетраедар 6.4).

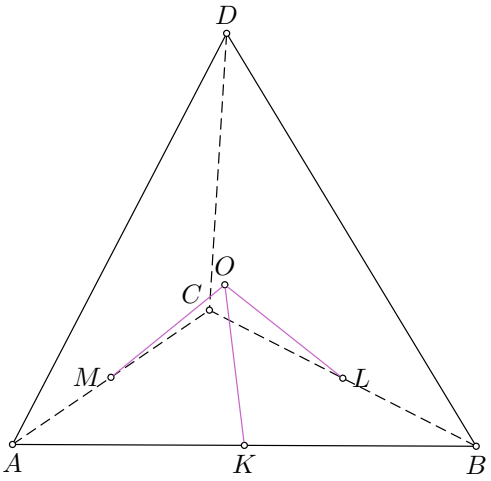
Када се изврши алгебризација коришћењем другог приступа, добија се 38 полинома који описују релације међу објектима, 6 полинома за које треба доказати да су једнаки нули (односно ти полиноми су полиноми који изражавају својство да се висине секу у једној тачки) и 41 променљива. Потом иде поступак поједностављивања и анализе добијених полинома. Број полинома који описују релације се смањи на 29, а и број променљивих се смањи на 29. Број полинома за које треба доказати да су нула је 9. Може бити чудно да се број полинома за које треба доказати да су нула увећао, али то увећање се добија због пресека правих. Наиме, тачке H_1, H_2 и H_3 (за које треба доказати да су једна иста тачка) се добијају као пресек правих, односно одговарајућих висина. Како за пресек правих треба доказати да оне нису мимоилазне, већ да се заиста и секу, за један или више полинома пресека је потребно доказати да они под датим условима су такође нула. О овом проблему је говорено раније 6.4. Полиноми добијени након поједностављивања су знатно краћи и најчешће се састоје од само два монома (у почетном скупу су били знатно комплекснији и у просеку су се састојали од 7 монома). Оба алгебарска доказивача (*GeoProver* и метод Гребнерових база имплементираних у систему *Mathematica*) су били успешни у доказивању да су свих девет полинома једнаки нули. Просечно време потребно за доказ у систему *GeoProver* је 0.0862 секунде.

Када се изврши алгебризација коришћењем првог приступа, добија се 34 полинома који описују релације међу објектима, 6 полинома за које треба доказати да су једнаки нули и 33 променљиве. Након поједностављивања добија се 22 полинома који описују релације међу објектима, 9 полинома за које треба доказати да су једнаки нули (исти разлог као и у другом приступу) и 25 различитих променљивих. Полиноми су за нијансу комплекснији него у првом приступу и у просеку имају 4 монома. Оба алгебарска доказивача (*GeoProver* и метод Гребнерових база имплементираних у систему *Mathematica*) су били успешни у доказивању да су свих девет полинома једнаки нули. Просечно време потребно за доказ у систему *GeoProver* је 0.1132 секунде.

	број полинома	број полинома доказа	просечан број монома	број промењливих	време
први приступ	22	9	4	25	$0.1132s$
други приступ	29	9	7	22	$0.0862s$

Табела 6.1: Упоредни приказ успешности алгебризације у односу на изабрани приступ

Пример једнаких углова. *Нека је $ABCD$ тетраедар и нека је тачка O центар описаног круга тетраедра $ABCD$. Нека су тачке K, L и M редом средине страница AB, BC и CA. Доказати да је $\angle KOL = \angle LOM = \angle MOK$.*



Слика 6.4: Тврђење о једнаким угловима између датих дужи

Тврђење се може записати на следећи начин:

```
make_tetrahedron A B C D

line_orth_plane_4points l1 A B C D
line_orth_plane_4points l2 A C D B

intersection_lines O l1 l2
```

```
midpoint K A B
midpoint L B C
midpoint M C A

equal_angles K O L L O M
```

Као и у претходном примеру тетраедар је у канонском положају према самој конструкцији, координате тачака су $A(0,0,0)$, $B(0,0,1)$, $C(x_1,x_2,0)$ и $S(x_3,x_4,x_5)$ при чему су x_1, x_2, x_3, x_4 и x_5 зависне променљиве и налазе се у полиномима који описују тетраедар (погледати у ранијем поглављу полиноме за тетраедар 6.4).

Коришћењем првог приступа добија се 31 полином који представља релације међу објектима, 1 полином за који је потребно доказати да је једнак нули и 36 променљивих. Након поједностављивања, добија се 4 полинома за које треба доказати да су једнаки нули (3 полинома служе као провера да ли се праве заиста секу, а један полином служи за проверу углова), 24 који описују релације међу објектима и 18 различитих променљивих. Међу добијеним полиномима, 15 полинома је прилично једноставно и састоје се од два монома. Ипак, преостали полиноми су комплексни и састоје се у просеку од осам монома. Метод Гребнерових база је био успешан у доказивању да су свих девет полинома једнаки нули. Систем *GeoProver* није био успешан и након 0.481 секунде пријавио је грешку да је досегао предвиђени меморијски лимит због полинома који има 3339 монома (енг. *Space limit exceeded in pseudo division. Obtained polynomial with 3339 terms*). Просечно време потребно за доказ да су полиноми који проверавају да ли има пресека једнаки нули у систему *GeoProver* је 0.377 секунду.

Коришћењем другог приступа, број добијених полинома је 28 и један полином за који је потребно доказати да је нула, број променљивих је 32. Након сређивања и анализе полинома пресека добија се 24 полинома и два полинома за које је потребно доказати да су нула и 24 различите променљиве. У овом приступу 18 полинома је било једноставно и састојало се из два монома, док је преосталих 6 полинома комплексније и састоје се у просеку од 8 монома. Оба алгебарска доказивача (*GeoProver* и метод Гребнерових база имплементираних у систему *Mathematica*) су били успешни у доказивању да су свих девет полинома једнаки нули. Време потребно за доказ да је полином који проверава да ли има пресека једнак нули је у систему *GeoProver* једнака

0.081 секунду, а време за доказ да је полином који проверава једнакост углова једнак нули у систему *GeoProver* је 0.835 секунде.

	број полинома	број полинома доказа	просечан број монома	број променљивих	време
први приступ	24	4	7.2	18	*Меморијски лимит*
други приступ	24	2	3.5	24	$0.835s$

Табела 6.2: Упоредни приказ успешности алгебризације у односу на изабрани приступ

Оно што се може приметити након ова два једноставна примера је да први приступ генерише мање променљивих, али да други приступ генерише једноставније полиноме. Како ће се показати даљим тестирањем, коришћеним алгебарским доказивачима више одговара када су полиноми једноставнији и зато су били успешнији када је коришћен други приступ.

Резултати методе Гребнерових база имплементираних у систему *Mathematica*. Метод Гребнерових база када је коришћен први приступ над 29 проблема је био успешан 23 пута и није доказао тврђења након 5 минута за 6 посматраних проблема.

Када је коришћен метод Гребнерових база али када је алгебризација рађена другим приступом, метод Гребнерових база је био успешан у свих 29 посматраних проблема.

Резултати система *GeoProver*. Систем *GeoProver* када је коришћен први приступ је био успешан само 13 пута. У различитим проблемима долазило је до различитих грешака због којих није успешно доказ завршен: достигнут је временски лимит (енг. *Time limit reached.*), достигнут је меморијски лимит (енг. *Space limit reached.*) или се догодила генерална грешка (енг. *General error occured.*). Последња грешка означава да доказивач није био у могућности да направи троугаони систем једначина. То на даље значи да је потребно додатно истражити ову ситуацију и потенцијално изменити улазне полиноме да би се ова грешка избегла. Ипак, за сада нисмо у могућности да овакво

понашање спречимо јер ако се током триангулације појави полином $0 = 0$ доказивач ће пријавити грешку. У почетном систему је лако могуће регистровати да ли такав полином постоји, али није лако могуће одредити да ли ће се у процесу триангулације такав полином створити јер је потребно испитати односе међу датим променљивима и задатим полиномима, што је доста комплексно питање.

Коришћењем другог приступа у алгебризацији, систем *GeoProver* је био нешто успешнији, од 29 посматраних проблема, систем је био успешан 22 пута. Грешке због којих није успео да заврши доказе су: временски лимит је досегнут или се догодила генерална грешка.

	GeoProver успех	*GeoProver* неуспех	*Гребнерове базе* успех	*Гребнерове базе* неуспех
први приступ	13	16	23	6
други приступ	22	7	29	0

Табела 6.3: Упоредни приказ успешности доказивача и приступа алгебризације

Закључак

Према нашем знању не постоји јавно доступан аутоматски доказивач за стереометрију. Алгебарски доказивачи (као што је Вуов метод или метод Гребнерових база) могу да доказују и тврђења у стереометрији, али је потребно та тврђења представити у полиномијалном облику.

У оквиру ове тезе изучавали смо како се може извршити алгебризација геометријских тврђења. Поступак алгебризације објеката и релација у стереометрији је могуће учинити на више начина и ми смо у овом раду представили два приступа. Први приступ уводи само координате тачака, а други приступ у једначине полинома уводи и координате правих и равни које учествују у геометријском тврђењу.

Поредили смо ова два приступа над истим скупом проблема и покретали смо два алгебарска доказивача, доказивач заснован на методи Гребнерових база и доказивач заснован на Вуовој методи. Један од првих закључака је да

Вуова метода захтева скуп полинома који се могу трансформисати у троугаони систем. Показало се да је понекад тај услов тешко испунити, а посебну потешкоћу је правила релација пресека две праве. Иако се овај проблем може решити, много погоднији за рад је метод Гребнерових база који овај захтев нема. Друга важна особина ове аутоматизације је била ефикасност доказивања у зависности од изабраног метода алгебризације. Тестирањем се показало да је боље када су полиноми једноставни без обзира на број полинома и број променљивих. Уколико су полиноми комплексни, онда су алгебарски доказивачи били мање успешни. Стога се други приступ показао као бољи јер се коришћењем тог приступа добијају једноставнији полиноми.

Једна од важних тема код алгебарских доказивача јесу услови недегенерисаности и у даљем раду би требало испитати како дати геометријску интерпретацију добијеним остацима полинома. Поред тога, пожељно је проширити систем тако да може да обухвати и обла тела (сфере, купе и ваљкове) и како одредити однос пресека правих са овим телима јер пресека може бити више.

У даљем раду циљ је направити заокружен систем за ефикасно визуелизовање и доказивање проблема у стереометрији. Као први корак би требало повезати направљени доказивач са динамичким геометријским софтвером који би истовремено успешно визуелизовао и доказивао геометријска тврђења. Још један занимљив корак би могао да буде трансформација геометријских тврђења са природног језика у термовску репрезентацију која би послужила за исцртавање и доказивање овог геометријског тврђења.

Глава 7

Закључци и даљи рад

7.1 Закључци

У овој тези представили смо формализацију Декартове координатне равни у оквиру система *Isabelle/HOL*. Дато је неколико различитих дефиниција Декартове координатне равни и доказано је да су све дефиниције еквивалентне. Дефиниције су преузете из стандардних уџбеника, али је било потребно подићи ниво ригорозности. Формално је доказано да Декартова координатна раван задовољава све аксиоме Тарског и већину аксиома Хилберта (укључујући и аксиому непрекидности). Показало се да, иако је већина тврђења једноставна, доказ тих тврђења захтева комплексна израчунавања и веома је захтеван за формализацију. Зато се веома често користи техника без губитка на општости која се заснива на изометријским трансформацијама.

У оквиру формализације хиперболичке геометрије представили смо формализацију геометрије проширене комплексне равни $\overline{\mathbb{C}}$ коришћењем комплексне пројективне равни, али и Риманове сфере. Формализовали смо аритметичке операције у $\overline{\mathbb{C}}$, размеру и дворазмеру, тетивну метрику у $\overline{\mathbb{C}}$, групу Мебијусових трансформација и њихово дејство на $\overline{\mathbb{C}}$, неке њене специјалне подгрупе (еуклидске сличности, ротације сфере, аутоморфизме диска), кругоправе и њихову везу са круговима и правама, тетивном метриком, Римановом сфером, јединственост кругоправи, дејство Мебијусових трансформација на кругоправе, типове и кардиналност скупа кругоправе, оријентисане кругоправе, однос између Мебијусових трансформација и оријентације, својство очувања угла након дејства Мебијусових трансформација итд. Кључан корак је био да се користи алгебарска репрезентација свих важних објеката (векто-

ра хомогених координата, матрица за Мебијусове трансформације, хермитске матрице за кругоправе итд.). Показало се да је алгебарски приступ далеко супериорнији у доказивању у односу на геометријски приступ. Релацију између у Поенкареовом диск моделу смо дефинисали коришћењем алгебарског приступа и показано је да шест аксиома Тарског важе у Поенкареовом диск моделу и да Еуклидова аксиома паралелности не важи.

Када је у питању формализација алгебарских доказивача у геометрији извршена је формализација алгебризације тврђења у планарној геометрији. Алгебарски доказивач заснован на методи Гребнерових база је већ формализован у оквиру система *Isabelle/HOL*. У оквиру ове тезе изучавали смо како се може извршити алгебризација геометријских тврђења задатих у тродимензионалном простору. Поступак алгебризације објеката и релација у стереометрији је могуће учинити на више начина и ми смо у овој тези представили два приступа. Први приступ уводи само координате тачака, а други приступ у једначине полинома уводи и координате правих и равни које учествују у геометријском тврђењу. Поредили смо ова два приступа над истим скупом проблема и покретали смо два алгебарска доказивача, доказивач заснован на методи Гребнерових база и доказивач заснован на Вуовој методи. Тестирањем се показало да је доказивање ефикасније када су полиноми једноставни без обзира на број полинома и број променљивих. Уколико су полиноми комплексни, онда су алгебарски доказивачи били мање успешни. Стога се други приступ у формализацији показао као бољи, јер се коришћењем тог приступа добијају једноставнији полиноми.

7.2 Даљи рад

У оквиру формализације аналитичке геометрије рад се може наставити на формализацији да наша дефиниција Декартове координатне равани задовољава све аксиоме Хилберта, односно увести појам угла, показати потребна својства и онда доказати аксиому комплетности. У нашем даљем раду планирамо да дефинишемо аналитичку геометрију у оквиру аксиоматизације Тарског или Хилберта. То би омогућило да докажемо категоричност и система аксиома Тарског и система аксиома Хилберта (и да докажемо да су сви модели изоморфни и еквивалентни Декартовој координатној равни). Поред бројних примена, комплексна геометрија је веома важна у физици и у многим

областима астрофизике се користе закључци и тврђења која управо потичу из хиперболичке геометрије. Било би интересантно испитивати и формализовати та тврђења и закључке и повезати са нашом постојећом формализацијом. Додатно, могуће је проширити анализу Мебијусових трансформација и показати да одређене класе ових трансформација задовољавају нека интересантна својства. У даљем раду на аутоматизацији стереометрије циљ је направити један заокружен систем за ефикасно визуелизовање и доказивање проблема у стереометрији. Као први корак би требало повезати направљени доказивач са динамичким геометријским софтвером који би истовремено успешно визуелизовао и доказивао геометријска тврђења. Други корак би могао да буде трансформација геометријских тврђења са природног језика у термовску репрезентацију која би послужила за исцртавање и доказивање овог геометријског тврђења. Додатно, ради прецизности и тачности, алгебарски доказивач за стереометрију би било могуће формализовати (као што је то урађено за планарну геометрију) и тако показати његову исправност.

Литература

[1] J Abbott, A Bigatti, and G Lagorio. Cocoa-5: A system for doing computations in commutative algebra, 2014.

[2] Mark Adams. Proof auditing formalised mathematics. *Journal of Formalized Reasoning*, 2016.

[3] Agda: An interactive proof editor. `http://agda.sf.net`.

[4] Michael Aschbacher and Stephen D Smith. *The classification of quasithin groups*, volume 2. American Mathematical Soc., 2004.

[5] Andrea Asperti, Wilmer Ricciotti, Claudio Sacerdoti Coen, and Enrico Tassi. The Matita interactive theorem prover. In *Automated Deduction–CADE-23*, pages 64–69. Springer, 2011.

[6] Jeremy Avigad, Edward Dean, and John Mumma. A formal system for Euclid's Elements. *The Review of Symbolic Logic*, 2(04):700–768, 2009.

[7] Jeremy Avigad, Kevin Donnelly, David Gray, and Paul Raff. A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic (TOCL)*, 9(1):2, 2007.

[8] Jeremy Avigad and John Harrison. Formally verified mathematics. *Communications of the ACM*, 57(4):66–75, 2014.

[9] Clemens Ballarin. Interpretation of locales in Isabelle: Theories and proof contexts. In *Mathematical Knowledge Management*, pages 31–43. Springer, 2006.

[10] Henk Barendregt, Wil Dekkers, and Richard Statman. *Lambda calculus with types*. Cambridge University Press, 2013.

[11] Bruno Barras. *Auto-validation d'un système de preuves avec familles inductives*. PhD thesis, 1999.

[12] Michael Beeson and Larry Wos. Finding proofs in Tarskian geometry. *to appear, The Journal of Automated Reasoning*, 2016.

[13] Jasmin Christian Blanchette, Lukas Bulwahn, and Tobias Nipkow. Automatic proof and disproof in Isabelle/HOL. In *Frontiers of Combining Systems*, pages 12–27. Springer, 2011.

[14] Francisco Botana and Miguel A Abánades. Automatic deduction in (dynamic) geometry: Loci computation. *Computational Geometry*, 47(1):75–89, 2014.

[15] Francisco Botana, Markus Hohenwarter, Predrag Janičić, Zoltán Kovács, Ivan Petrović, Tomás Recio, and Simon Weitzhofer. Automated theorem proving in GeoGebra: current achievements. *Journal of Automated Reasoning*, 55(1):39–59, 2015.

[16] Francisco Botana, Markus Hohenwarter, Predrag Janičić, Zoltán Kovács, Ivan Petrović, Tomás Recio, and Simon Weitzhofer. Automated theorem proving in GeoGebra: Current achievements. *Journal of Automated Reasoning*, 55(1):39–59, 2015.

[17] Francisco Botana and José L Valcarce. A dynamic–symbolic interface for geometric theorem discovery. *Computers & Education*, 38(1):21–35, 2002.

[18] Pierre Boutry, Gabriel Braun, and Julien Narboux. From Tarski to Descartes: Formalization of the arithmetization of Euclidean geometry. In *SCSS 2016 The 7th International Symposium on Symbolic Computation in Software Science*, 2016.

[19] Pierre Boutry, Julien Narboux, and Pascal Schreck. Parallel postulates and decidability of intersection of lines: a mechanized study within Tarski's system of geometry. 2015.

[20] Pierre Boutry, Julien Narboux, and Pascal Schreck. A reflexive tactic for automated generation of proofs of incidence to an affine variety. 2015.

[21] R. S. Boyer and J S. Moore. *A Computational Logic Handbook*. Academic Press, New York, 1988.

[22] Gabriel Braun and Julien Narboux. From Tarski to Hilbert. In *Automated Deduction in Geometry*, pages 89–109. Springer, 2012.

[23] Gabriel Braun and Julien Narboux. A synthetic proof of Pappus' theorem in Tarski's geometry. 2015.

[24] Bruno Buchberger. Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal. *Journal of symbolic computation*, 41(3):475–511, 2006.

[25] Bruno Buchberger and Franz Winkler. *Gröbner bases and applications*, volume 251. Cambridge University Press, 1998.

[26] XueFeng Chen and DingKang Wang. The projection of quasi variety and its application on geometric theorem proving and formula deduction. In *International Workshop on Automated Deduction in Geometry*, pages 21–30. Springer, 2002.

[27] Shang-Ching Chou. Proving elementary geometry theorems using Wu's algorithm. Master's thesis, University of Texas at Austin, 1984.

[28] Shang-Ching Chou. *Mechanical geometry theorem proving*, volume 41. Springer Science & Business Media, 1988.

[29] Shang-Ching Chou and Xiao-Shan Gao. Automated reasoning in geometry. *Handbook of automated reasoning*, 1:707–749, 2001.

[30] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated production of traditional proofs for constructive geometry theorems. In *Logic in Computer Science, 1993. LICS'93., Proceedings of Eighth Annual IEEE Symposium on*, pages 48–56. IEEE, 1993.

[31] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. *Machine proofs in geometry: Automated production of readable proofs for geometry theorems*, volume 6. World Scientific, 1994.

[32] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated production of traditional proofs in solid geometry. *Journal of Automated Reasoning*, 14(2):257–291, 1995.

[33] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants. *Journal of Automated Reasoning*, 17(3):325–347, 1996.

[34] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. An introduction to geometry expert. In *Automated Deduction—Cade-13*, pages 235–239. Springer, 1996.

[35] Shang-Ching Chou, William F Schelter, and Jin-Gen Yang. Characteristic sets and Gröbner bases in geometry theorem proving. *Resolution of equations in algebraic structures*, 2:33–91, 1987.

[36] Alonzo Church. A formulation of the simple theory of types. *The journal of symbolic logic*, 5(02):56–68, 1940.

[37] Alonzo Church. *The calculi of lambda-conversion*. Number 6. Princeton University Press, 1941.

[38] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T. Saski, and S. F. Smith. *Implementing mathematics with the Nuprl proof development system*. Prentice Hall, 1986.

[39] The Coq proof assistant. `http://coq.inria.fr/`.

[40] Pierre Corbineau. A declarative language for the Coq proof assistant. In *Types for Proofs and Programs*, pages 69–84. Springer, 2007.

[41] David Cox, John Little, and Donal O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.

[42] Luís Cruz-Filipe. A constructive formalization of the fundamental theorem of calculus. In *Types for Proofs and Programs*, pages 108–126. Springer, 2002.

[43] Haskell B Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences*, 20(11):584–590, 1934.

[44] N. G. de Bruijn. The mathematical language AUTOMATH. volume 25 of *Lecture Notes in Mathematics*, pages 29–61. Springer-Verlag, Berlin, 1970.

[45] Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck. Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry. In *Automated Deduction in Geometry*, pages 306–323. Springer, 2001.

[46] Andreas Dolzmann, Thomas Sturm, and Volker Weispfenning. A new approach for automatic theorem proving in real geometry. *Journal of Automated Reasoning*, 21(3):357–380, 1998.

[47] Jean Duprat. Une axiomatique de la géométrie plane en Coq. *Actes des JFLA*, pages 123–136, 2008.

[48] Frederic B. Fitch. *Symbolic Logic: An Introduction.* The Ronald Press Company, New York, 1952.

[49] Laurent Fuchs and Laurent Théry. A formalization of grassmann-cayley algebra in COQ and its application to theorem proving in projective geometry. In *Automated Deduction in Geometry*, pages 51–67. Springer, 2011.

[50] Xiao-Shan Gao and Shang-Ching Chou. Computations with parametric equations. In *Proceedings of the 1991 international symposium on Symbolic and algebraic computation*, pages 122–127. ACM, 1991.

[51] Xiaoshan Gao. Transcendental functions and mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning*, 6(4):403–417, 1990.

[52] Herbert Gelernter. Realization of a geometry theorem proving machine. In *IFIP Congress*, pages 273–281, 1959.

[53] Jean-David Génevaux, Julien Narboux, and Pascal Schreck. Formalization of Wu's simple method in Coq. In *Certified Programs and Proofs*, pages 71–86. Springer, 2011.

[54] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische zeitschrift*, 39(1):176–210, 1935.

[55] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39(1):405–431, 1935.

[56] Herman Geuvers. Proof assistants: History, ideas and future. *Sadhana*, 34(1):3–25, 2009.

[57] Herman Geuvers, Freek Wiedijk, and Jan Zwanenburg. A constructive proof of the Fundamental theorem of algebra without using the rationals. In *Types for Proofs and Programs*, pages 96–111. Springer, 2000.

[58] Jean-Yves Girard. *Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur*. PhD thesis, PhD thesis, Université Paris VII, 1972.

[59] Jean-Yves Girard. The system F of variable types, fifteen years later. *Theoretical computer science*, 45:159–192, 1986.

[60] Jean-Yves Girard, Yves Lafont, and Paul Taylor. Proofs and types, volume 7 of Cambridge tracts in theoretical computer science, 1989.

[61] Georges Gonthier. Formal proof–the four-color theorem. *Notices of the AMS*, 55(11):1382–1393, 2008.

[62] Georges Gonthier. Formal proof–the four-color theorem. *Notices of the AMS*, 55(11):1382–1393, 2008.

[63] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, et al. A machine-checked proof of the odd order theorem. In *International Conference on Interactive Theorem Proving*, pages 163–179. Springer, 2013.

[64] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, et al. A machine-checked proof of the odd order theorem. In *Interactive Theorem Proving*, pages 163–179. Springer, 2013.

[65] Georges Gonthier, Assia Mahboubi, and Enrico Tassi. *A small scale reflection extension for the Coq system*. PhD thesis, Inria Saclay Ile de France, 2014.

[66] Mike Gordon. From LCF to HOL: a short history. In *Proof, Language, and Interaction*, pages 169–186, 2000.

[67] Benjamin Grégoire, Loïc Pottier, and Laurent Théry. Proof certificates for algebra and their application to automatic geometry theorem proving. In *Automated Deduction in Geometry*, pages 42–59. Springer, 2011.

[68] Jérémie Gressier. Geometrix IV. `http://geometrix.free.fr/`, 2013.

[69] Frédérique Guilhot. Formalisation en Coq et visualisation d'un cours de géométrie pour le lycée. *Technique et Science informatiques*, 24(9):1113–1138, 2005.

[70] Florian Haftmann, Alexander Krauss, Ondrej Kuncar, and Tobias Nipkow. Data refinement in Isabelle/HOL. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 100–115. Springer, 2013.

[71] Florian Haftmann and Makarius Wenzel. Constructive type classes in Isabelle. In *International Workshop on Types for Proofs and Programs*, pages 160–174. Springer, 2006.

[72] Thomas C Hales. Introduction to the Flyspeck project. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2006.

[73] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, 1993.

[74] Robert Harper, David MacQueen, and Robin Milner. *Standard ML*. University of Edinburgh. Department of Computer Science. Laboratory for Foundations of Computer Science, 1986.

[75] J. Harrison. HOL Light: A tutorial introduction. *Lecture Notes in Computer Science*, 1166:265–269, 1996.

[76] John Harrison. A HOL theory of Euclidean space. In *Theorem proving in higher order logics*, pages 114–129. Springer, 2005.

[77] John Harrison. HOL Light: An overview. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics, 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*, volume 5674 of *Lecture Notes in Computer Science*, pages 60–66. Springer, 2009.

[78] John Harrison. Without loss of generality. In *Theorem Proving in Higher Order Logics*, pages 43–59. Springer, 2009.

[79] John Harrison. *Theorem proving with the real numbers*. Springer Science & Business Media, 2012.

[80] John Harrison. The HOL Light theory of Euclidean space. *Journal of Automated Reasoning*, 50(2):173–190, 2013.

[81] David Hilbert. *Grundlagen der geometrie*. Springer-Verlag, 2013.

[82] E Hille. Analytic function theory (Chelsea, New York). *Vol. II*, page 375, 1973.

[83] Paul Hudak, Simon Peyton Jones, Philip Wadler, Brian Boutel, Jon Fairbairn, Joseph Fasel, María M Guzmán, Kevin Hammond, John Hughes, Thomas Johnsson, et al. Report on the programming language Haskell: a non-strict, purely functional language version 1.2. *ACM SigPlan notices*, 27(5):1–164, 1992.

[84] Brian Huffman and Ondřej Kunčar. Lifting and Transfer: A modular design for quotients in Isabelle/HOL. In *Certified Programs and Proofs*, pages 131–146. Springer, 2013.

[85] Predrag Janicic. Zbirka zadataka iz geometrije. *Skripta Internacional, Beograd,*, 1997.

[86] Predrag Janičić. Geometry constructions language. *Journal of Automated Reasoning*, 44(1-2):3–24, 2010.

[87] Predrag Janičić, Julien Narboux, and Pedro Quaresma. The area method. *Journal of Automated Reasoning*, 48(4):489–532, 2012.

[88] Stanisław Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1:5–32, 1934. Reprinted in: S. McCall (ed.), Polish Logic 1920–1939, Clarendon Press, Oxford, pp. 232–258.

[89] Benthem Jutting and Bert van LS. *Checking Landau's' Grundlagen'in the AUTOMATH system*. PhD thesis, Mathematisch Centrum, 1977.

[90] Gilles Kahn. Constructive geometry according to Jan von Plato. *Coq contribution. Coq*, 5:10, 1995.

[91] Cezary Kaliszyk and Christian Urban. Quotients revisited for Isabelle/HOL. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 1639–1644. ACM, 2011.

[92] Deepak Kapur and Hoi K Wan. Refutational proofs of geometry theorems via characteristic set computation. In *Proceedings of the international symposium on Symbolic and algebraic computation*, pages 277–284. ACM, 1990.

[93] Gerwin Klein, June Andronick, Kevin Elphinstone, Toby Murray, Thomas Sewell, Rafal Kolanski, and Gernot Heiser. Comprehensive formal verification of an OS microkernel. *ACM Transactions on Computer Systems (TOCS)*, 32(1):2, 2014.

[94] Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, et al. sel4: Formal verification of an os kernel. In *Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, pages 207–220. ACM, 2009.

[95] Ulrich Kortenkamp and Jürgen Richter-Gebert. Using automatic theorem proving to improve the usability of geometry software. In *Proceedings of MathUI*, volume 2004, 2004.

[96] Alexander Krauss. Defining recursive functions in Isabelle/HOL, 2008.

[97] Saul A Kripke. Semantical analysis of intuitionistic logic I. *Studies in Logic and the Foundations of Mathematics*, 40:92–130, 1965.

[98] Bernhard Kutzler and Sabine Stifter. On the application of Buchberger's algorithm to automated geometry theorem proving. *Journal of Symbolic Computation*, 2(4):389–397, 1986.

[99] Edmund Landau. *Grundlagen der Analysis:(das Rechnen mit ganzen, rationalen, irrationalen, komplexen Zahlen)*. Chelsea Pub. Co., 1960.

[100] Xavier Leroy. The OCaml programming language. *Online: http://caml. inria. fr/ocaml/index. en. html*, 1998.

[101] Xavier Leroy. Formal church1940formulationverification of a realistic compiler. *Communications of the ACM*, 52(7):107–115, 2009.

[102] Xavier Leroy. A formally verified compiler back-end. *Journal of Automated Reasoning*, 43(4):363–446, 2009.

[103] Xavier Leroy. A formally verified compiler back-end. *Journal of Automated Reasoning*, 43(4):363–446, 2009.

[104] Xavier Leroy, Damien Doligez, Alain Frisch, Jacques Garrigue, Didier Rémy, and Jérôme Vouillon. The OCaml system release 4.02. *Institut National de Recherche en Informatique et en Automatique*, 2014.

[105] Pierre Letouzey. A new extraction for Coq. In *International Workshop on Types for Proofs and Programs*, pages 200–219. Springer, 2002.

[106] NJ Lord. A method for vector proofs in geometry. *Mathematics Magazine*, 58(2):84–89, 1985.

[107] Zhaohui Luo and Robert Pollack. The LEGO proof development system: A user's manual. Technical Report ECS-LFCS-92-211, University of Edinburgh, May 1992.

[108] Nicolas Magaud, Julien Narboux, and Pascal Schreck. Formalizing projective plane geometry in Coq. In *Automated Deduction in Geometry*, pages 141–162. Springer, 2011.

[109] Nicolas Magaud, Julien Narboux, and Pascal Schreck. A case study in formalizing projective geometry in Coq: Desargues theorem. *Computational Geometry*, 45(8):406–424, 2012.

[110] Lena Magnusson and Bengt Nordström. The ALF proof editor and its proof engine. In Henk Barendregt and Tobias Nipkow, editors, *Types for Proofs and Programs*, pages 213–237. Springer-Verlag LNCS 806, 1994.

[111] Timothy James McKenzie Makarios. A mechanical verification of the independence of Tarski's Euclidean axiom. 2012.

[112] Filip Maric. A survey of interactive theorem proving.

[113] Filip Marić, Ivan Petrović, Danijela Petrović, and Predrag Janičić. Formalization and implementation of algebraic methods in geometry. *arXiv preprint arXiv:1202.4831*, 2012.

[114] Vesna Marinković, Predrag Janičić, and Pascal Schreck. Computer theorem proving for verifiable solving of geometric construction problems. In *Automated Deduction in Geometry*, pages 72–93. Springer, 2014.

[115] P. Martin Löf. *Intuitionistic Type Theory*. Studies in Proof Theory, Bibliopolis, Napoli, 1984.

[116] Per Martin-Löf. Intuitionistic type theory. *Naples: Bibliopolis*, 76, 1984.

[117] Miodrag Mateljević. *Kompleksne funkcije 1 & 2*. Društvo matematičara Srbije, 2006.

[118] Conor McBride. Epigram: Practical programming with dependent types. In *Advanced Functional Programming*, pages 130–170. Springer, 2004.

[119] John McCarthy, Steve Russell, Timothy P Hart, Mike Levin, AutoLISP Arc, and Common Lisp Clojure. Lisp programming language, 1985.

[120] John D McCharen, Ross A Overbeek, and LAWRENCE T WOS. Problems and experiments for and with automated theorem-proving programs. In *The Collected Works of Larry Wos: (In 2 Volumes) Volume I: Exploring the Power of Automated Reasoning Volume II: Applying Automated Reasoning to Puzzles, Problems, and Open Questions*, pages 166–196. 2000.

[121] Laura I Meikle and Jacques D Fleuriot. Formalizing Hilbert's Grundlagen in Isabelle/Isar. In *Theorem proving in higher order logics*, pages 319–334. Springer, 2003.

[122] Robert Milewski. Fundamental theorem of algebra1. 2001.

[123] Robin Milner. Implementation and applications of Scott's logic for computable functions. In *ACM sigplan notices*, volume 7, pages 1–6. ACM, 1972.

[124] John C Mitchell. *Foundations for programming languages*, volume 1. MIT press Cambridge, 1996.

[125] M.J.C. Gordon and T.F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, 1993.

[126] César A Munoz and Gilles Dowek. Hybrid verification of an air traffic operational concept. *lateral*, 5000(4000):3000, 2005.

[127] Julien Narboux. Mechanical theorem proving in Tarski's geometry. In *Automated Deduction in Geometry*, pages 139–156. Springer, 2007.

[128] Julien Narboux. Mechanical theorem proving in Tarski's geometry. In *Automated Deduction in Geometry*, pages 139–156. Springer, 2007.

[129] Julien Narboux and David Braun. Towards a certified version of the Encyclopedia of triangle centers. 2015.

[130] Tristan Needham. *Visual complex analysis*. Oxford University Press, 1998.

[131] Tobias Nipkow, Lawrence C Paulson, and Markus Wenzel. *Isabelle/HOL: a proof assistant for higher-order logic*, volume 2283. Springer Science & Business Media, 2002.

[132] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. Oxford University Press, 1990.

[133] Ulf Norell. *Towards a practical programming language based on dependent type theory*, volume 32. Citeseer, 2007.

[134] Sam Owre, John Rushby, N Shankar, et al. Pvs specification and verification system. *URL: pvs. csl. sri. com*, 2001.

[135] Erik Palmgren. Semantics of intuitionistic propositional logic. *Lecture Notes for Applied Logic*, 2009.

[136] L. C. Paulson. Natural deduction as higher-order resolution. *Journal of Logic Programming*, 3:237–258, 1988.

[137] Lawrence C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–397, 1989.

[138] Lawrence C Paulson. *Logic and computation: interactive proof with Cambridge LCF*, volume 2. Cambridge University Press, 1990.

[139] Tuan-Minh Pham, Yves Bertot, and Julien Narboux. A Coq–based library for interactive and automated theorem proving in plane geometry. In *Computational Science and Its Applications-ICCSA 2011*, pages 368–383. Springer, 2011.

[140] Loic Pottier. Connecting Gr\„ obner bases programs with Coq to do proofs in algebra, geometry and arithmetics. *arXiv preprint arXiv:1007.3615*, 2010.

[141] Loic Pottier. Connecting Gr\„ obner bases programs with Coq to do proofs in algebra, geometry and arithmetics. *arXiv preprint arXiv:1007.3615*, 2010.

[142] Art Quaife. *Automated development of fundamental mathematical theories*. JSTOR, 1992.

[143] David L. Rager, Warren A. Hunt Jr., and Matt Kaufmann. A parallelized theorem prover for a logic with parallel execution. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 435–450. Springer, 2013.

[144] William Richter. A minimal version of Hilbert's axioms for plane geometry.

[145] Jürgen Richter-Gebert. *Foundations of Dynamic Geometry*. PhD thesis, SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH, 1999.

[146] Joseph Fels Ritt. *Differential algebra*, volume 33. American Mathematical Soc., 1950.

[147] Wolfram Schwabhäuser, Wanda Szmielew, and Alfred Tarski. *Metamathematische methoden in der geometrie*. Springer-Verlag, 2013.

[148] Hans Schwerdtfeger. *Geometry of complex numbers: circle geometry, Moebius transformation, non-euclidean geometry*. Courier Corporation, 1979.

[149] Phil Scott. Mechanising Hilbert's foundations of geometry in Isabelle. *Master's thesis, University of Edinburgh*, 2008.

[150] WANG DONG-MING HU SEN. A mechanical proving system for constructible theorems in elementary geometry. 1987.

[151] Natarajan Shankar. *Metamathematics, machines and Gödel's proof*. Number 38. Cambridge University Press, 1997.

[152] Changpeng Shao, Hongbo Li, and Lei Huang. Challenging theorem provers with Mathematical Olympiad problems in solid geometry. *Mathematics in Computer Science*, 10(1):75–96, 2016.

[153] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry–Howard isomorphism*, volume 149. Elsevier, 2006.

[154] Michael Sperber, R Kent Dybvig, Matthew Flatt, Anton van Straaten, Richard Kelsey, William Clinger, and Jonathan Rees. Revised6 report on the algorithmic language Scheme (Libraries), 2007.

[155] Christian Sternagel and René Thiemann. Executable multivariate polynomials. 2013.

[156] Sabine Stifter. Geometry theorem proving in vector spaces by means of Gröbner bases. In *Proceedings of the 1993 international symposium on Symbolic and algebraic computation*, pages 301–310. ACM, 1993.

[157] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, 1, 1989.

[158] Andrzej Trybulec and Howard Blair. Computer assisted reasoning with MIZAR. In Aravind Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 26–28, Los Angeles, CA, August 1985. Morgan Kaufmann.

[159] Sana Stojanović Đurđević, Julien Narboux, and Predrag Janičić. Automated generation of machine verifiable and readable proofs: A case study of Tarski's geometry. *Annals of Mathematics and Artificial Intelligence*, 74(3-4):249–269, 2015.

[160] Jan von Plato. The axioms of constructive geometry. *Annals of pure and applied logic*, 76(2):169–200, 1995.

[161] Dongming Wang. Elimination procedures for mechanical theorem proving in geometry. *Annals of Mathematics and Artificial Intelligence*, 13(1-2):1–24, 1995.

[162] Dongming Wang. Decomposing polynomial systems into simple systems. *Journal of Symbolic Computation*, 25(3):295–314, 1998.

[163] Dongming Wang. Geother 1.1: Handling and proving geometric theorems automatically. In *Automated Deduction in Geometry*, pages 194–215. Springer, 2002.

[164] Wu Wen-Tsun. Mechanical theorem proving of differential geometries and some of its applications in mechanics. *Journal of Automated Reasoning*, 7(2):171–191, 1991.

[165] Wu Wen-Tsun. On a finiteness theorem about optimization problems. Technical report, 1992.

[166] Markus Wenzel et al. *Isabelle/Isar—a versatile environment for human-readable formal proof documents*. PhD thesis, PhD thesis, Institut für Informatik, Technische Universität München, 2002. http://tumb1. biblio. tu-muenchen. de/publ/diss/in/2002/wenzel. html, 2002.

[167] Freek Wiedijk. Mizar light for HOL Light. In *Theorem Proving in Higher Order Logics*, pages 378–393. Springer, 2001.

[168] Sean Wilson and Jacques D Fleuriot. Combining dynamic geometry, automated geometry theorem proving and diagrammatic proofs. In *Workshop on User Interfaces for Theorem Provers (UITP)*, 2005.

[169] Inc Wolfram Research. Mathematica, 2008.

[170] Wen-tsün Wu. On the decision problem and the mechanization of theorem-proving in elementary geometry. *Scientia Sinica*, 21(2):159–172, 1978.

[171] Wenjun Wu and Xiaoshan Gao. Mathematics mechanization and applications after thirty years. *Frontiers of Computer Science in China*, 1(1):1–8, 2007.

[172] L Yang, X Gao, S Chou, and Z Zhang. Automated proving and discovering of theorems in non-euclidean geometries. *Proceedings of Automated Deduction in Geometry (ADG98), Lecture Notes in Artificial Intelligence*, 1360:171–188, 1998.

[173] Zheng Ye, Shang-Ching Chou, and Xiao-Shan Gao. An introduction to java geometry expert. In *Automated Deduction in Geometry*, pages 189–195. Springer, 2008.

[174] Zbirka zadataka iz geometrije prostora za pripremu prijemnog ispita na Arhitektonskom fakultetu. Arhitektonska tehnicka skola, Beograd. Skripta, 2001.

Биографија аутора

Данијела Симић (рођена Петровић) рођена је 26.09.1986. године у Ваљеву. Основну школу „Жикица Јовановић Шпанац" и „Ваљевску гимназију" завршила је као ђак генерације и носилац Вукове дипломе. Током средње и основне школе била је ученик многих такмичења из математике, физике, програмирања и освојила је бројне награде од којих су најважније трећа награда на савезном такмичењу из физике у четвртој години средње школе, као и трећа награда на републичком такмичењу из програмирања у четвртом разреду средње школе.

Године 2005. уписала је Математички факултет, универзитета у Београду, смер Рачунарство и Информатика. Студије је завршила 2009. године са просечном оценом 9.86. Била је стипендиста Министарства за науку и технологију. Докторске академске студије на смеру Информатика уписала је октобра 2009. године. Све испите предвиђене планом студија положила је са оценом 10. Учествовала је у организацији више научних скупова у земљи које је организовала *Аргο γруūа* Математичког факултета, чији је члан од 2010. године. До сада је у два циклуса била учесник научних пројеката које је финансирало Министарство просвете, науке и технолошког развоја Владе Републике Србије.

Основна област интересовања јој је аутоматско резоновање, са акцентом на аутоматско и интерактивно доказивање у геометрији. Имала је излагања на неколико конференција и радионица у земљи и иностранству.

Од октобра 2009. године била је запослена као сарадник у настави, а од октобра 2011. године као асистент на Катедри за рачунарство и информатику Математичког факултета Универзитета у Београду. Током досадашњег рада на Математичком факултету држала је вежбе из 6 предмета.

Прилог 1.

Изјава о ауторству

Потписани-а _____

број индекса _____

Изјављујем

да је докторска дисертација под насловом

_____

_____

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, _____

_____

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора _____

Број индекса _____

Студијски програм _____

Наслов рада _____

Ментор _____

Потписани/а _____

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, _____

_____

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

_____

_____

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, _____

_____