پیشتر، از همیاری و دقت نظر اعضای کمیته‌ی دفاع، آقایان دکتر *امیر دانشگر* و دکتر *مرتضی منیری* و نیز از پیگیری دلسوزانه‌ی جناب آقای دکتر *حمیدرضا فنایی* سپاسگزارم.

و این پایان‌نامه را - هر چند ناچیز - تقدیم می کنم به استادم دکتر *محمد اردشیر* و روان پاک مادر بزرگوارش که بی کم و کاست پیام آور اندیشه‌ی شهودگرایانه در ایران را در دامان-اش پروراند.

نیما دارابی
تهران، اَمرداد ۱۳۸۶

دانشگاه صنعتی شریف

دانشکده ریاضی

پایان نامه کارشناسی ارشد

بررسی روش‌های استنتاج خودکار

در منطق های موجهات

نیما دارابی

استاد راهنما: دکتر محمد اردشیر

تابستان ۱۳۸۶

Sharif University of Technology

Faculty of Mathematics

M.S. Thesis

Automated Theorem Proving Methods in Modal Logics

Nima Darabi

Supervisor: Dr. Mohammad Ardeshir

Simmer 2007

فهرست

◆

پیشگفتار

علوم رایانه و از جمله‌ی آن هوش مصنوعی به سختی با ایده‌های منطقی گره خورده است. سه کاربرد منطق را در هوش مصنوعی می توان بر می‌شمارد: به عنوان ابزاری برای پردازش[۱]، به عنوان پایه ای برای بازنمائی آگاهی[۲]، به عنوان یک زبان برنامه سازی[۳] [1].

منطق دانش مطالعه‌ی اصول استدلال کردن است. استدلال های صوری را بر مبنای قوانین منطق بررسی می کنند. ریشه های این زمینه به اواخر قرن نوزده بر می‌گردد، یعنی زمانی که فرگه کتاب[۴]-اش را نگاشت. این نخستین کوشش جامع برای بنیاد نهادن یک زبان صوری بود که بتواند زیربنایی برای ریاضیات باشد. افسوس که راسل پارادوکسی یافت که نشان می‌داد سیستم فرگه ناسازگار است: ارزش هر گزاره می‌تواند درون خودش محاسبه شود. پس از آن راسل سیستم خود را بر مبنای نظریه

---

[1] A tool for proccessing

[2] A basis for knowledge representation

[3] A programming Language

[4] Baiffsschrift

ی انواع[5] بسط داد و با همکاری وایتهد[6] در کتاب اصول ریاضیات[7] نشان داد که چگونه می توان آن را زیربنای منطق قرار داد. هیلبرت[8] یک جایگزین ساده تر را به دست آورد: حساب محمولات[9] . پس از او فرمول بندی گنتزن[10] از حساب مرتبه ی اول در سیستمی به فرم استنتاج طبیعی[11] نقطه ی عطفی در این زمینه بود. در این روش معنای هر رابط منطقی بر اساس قوانین استنتاج بیان می شود. او همچنین ابزار فنی حساب رشته‌ها[12] را پدید آورد و نشان داد که آن هم ارز دستگاه استناج طبیعی است. بسیاری از روش های استنتاج خودکار بر پایه ی همین حساب استوار شده اند. می توانیم از جمله ی این روش ها به آنها اشاره کنیم که عقبگرد کار می کنند و از قضیه ی پیشنهاد شده به اصول موضوعه می رسند یا آنکه از اصول موضوعه جلوگرد می کنند و قضیه را ثابت می کنند. گرچه بعدها نشان داده شده است که ترکیبی از این دو روش بهتر کار می کند. انواع این روش ها را در این نوشتار بررسی خواهیم کرد.

اما پرسش دیگری که در این میان خودنمایی می کند این است: در کدام منطق ما می خواهیم استدلال کنیم. اینکه دستگاه فکری ما چه باشد، در گزاره های مورد قبول ما تأثیر می گذارد؛ اینکه چه اصولی را بپذیریم و یا نه. پس هر دستگاه اصل موضوعی باید روش های منحصر به خود را داشته باشد و همین است که مبحث استدلال خودکار را تا به این حد گسترده کرده و محققین زیادی را بر پرسش‌های بی‌شمار خود گمارده است.

[5] Type Theory

[6] Whitehead

[7] Principia Mathematica

[8] Hilbert

[9] Predicate Calculus

[10] Gentzen

[11] Natural Deduction

[12] Sequent Calculus

موضوع این پایان‌نامه:

در این پایان‌نامه روش‌های استدلال خودکار را برای منطق‌های موجهات بررسی کرده ایم. همچنین روش های ارائه شده پیاده سازی شده اند. در پیاده سازی روش هیلبرتی ایده هایی برای سرعت بخشیدن به جستجوی برهان به کار گرفته شده که در فصل دوم شرح داده شده اند. منبع اصلی روش‌ها در این پروژه برای منطق های موجهات [16] و [1]، و برای منطق های موجهات شهودگرایانه [1] بوده است.

در فصل نخست این پایان‌نامه به معرفی منطق‌های موجهات ارائه شده خواهیم پرداخت. در فصل دوم *دستگاه استنتاج اصل موضوعی هیلبرتی*[۱۳] را معرفی خواهیم کرد. در این فصل ایده‌هایی را ذکر خواهیم کرد که برای پیاده‌سازی و نیز کارآمدتر کردن این روش‌ها (چه در حالت کلی و چه در حالت خاص برای منطق‌های موجهات) استفاده کرده ایم. در فصل سوم روش *تابلو*[۱۴] برای این نوع منطق بسط داده شده و پیاده‌سازی شده‌اند. فصل چهارم نیز به روش *رشته‌های گنتزن*[۱۵] و کاربرد تعمیم یافته‌ی آن برای این نوع منطق‌ها اختصاص یافته است. فصل پنجم نیز شامل معرفی منطق های موجهات شهودگرایانه و درآمدی بر روش‌های استنتاج خودکار در آنها می‌باشد.

---

[13] Hilbert deduction system

[14] Tableau

[15] Gentzen Sequent

۱

معرفی منطق های موجهات

از میان انواع منطق‌ها، منطق موجهات[16] به هر منطقی گفته می‌شود که قید و شرط‌ها را سامان دهد. یک قید[17] به هر واژه یا کلمه ای گفته می‌شود که بتواند به یک جمله‌ی داده شده اعمال شود و جمله‌ی جدیدی را پدید آورد به نحوی که دربردارنده‌ی ادعایی درباره‌ی وضعیت[18] صحت آن جمله باشد: کی، کجا، چگونه و در چه شرایطی برقرار است.

منطق های موجهات می‌توانند به عنوان منطقی برای حقایق *لازم* و *محتمل* به کار روند: "*باید چترت را ببری، چون شاید فردا ببارد*". نیز شرط‌ها را می توان به صورت قیدهای زبان طبیعی در نظر گرفت. در اینجا است که منطق های گوناگونی از دل منطق موجهات سر در می آورند:

---

16 Modal Logic

17 Modality

18 Mode

منطق زمان: *در گذشته ...، هم اکنون ...*

منطق شناختی: *دانش عامه بر این است که ...*

منطق تکلیف: *قانونی است که ...، مجاز است که ...، الزامی است که ...*

منطق doxastic: *این باور وجود دارد که...*

منطق هندسی: *در این ناحیه ...*

در این فصل به ترتیب نحو[19] و معنا[20] در منطق موجهات را ارائه خواهیم کرد. از آنجا که در این پایان نامه روش‌های استنتاج خودکار در منطق‌های شهودگرایانه نیز مورد بررسی اجمالی قرار گرفته اند، فصل ۵ را به منطق‌های شهودگرایانه و منطق های موجهاتی که بر پایه ی آنها ساخته می شود یعنی منطق موجهات شهودگرایانه اختصاص داده ایم. برای خوانایی بیشترِ کدهای نوشته شده در پیاده سازی این پایان نامه، این تعاریف را هماهنگ با اشیاء به کار رفته در نرم افزار اتخاذ کرده ایم. این نرم افزار در لوح فشرده ای بر این پایان نامه پیوست شده است.

این منطق را چندان که توصیف اش در پی می آید PL [21] می نامیم.

ابتکار تعریف منطق‌های موجهات در واقع به اوایل قرن گذشته باز می گردد. در ۱۹۱۸ بود که لویس[22] از محدودیت عملگرهای منطق کلاسیک گذر کرد و منطق جدیدی پدید آورد که در آن از عملگر الزام ( ) استفاده شده بود. این عملگر بنا است که مفهوم ″باید″ را در زبان طبیعی معنا دهد. عملگر دیگری که می‌تواند نشانگر ″شاید″ باشد، عملگر امکان (◊) است. $\Diamond P$ می‌تواند بر اساس عملگر الزام به صورت $\neg \ \neg P$ نیز تعریف شود. در این حد، هنوز شهود ما می‌تواند هم ارزی $\neg \ \neg P$ و $\Diamond P$ را

[19] Syntax

[20] Semantics

[21] Propositional Logic

[22] Lewis

تأیید کند. "چنین نیست که باید نروژ زمستان های سردی داشته باشد"؛ ما از این جمله می‌فهمیم که "شاید نروژ زمستان‌های گرمی داشته باشد".

جلوتر به معناشناسی منطق موجهات بیشتر می پردازیم. در اینجا ذکر این نکته شهودا مفید است که p به شیوه‌های گوناگون تعبیر می‌شود. می‌توان این منطق را منطق دانش دانست و P را چنین خواند که "می دانیم که P". این در واقع به این معنا است که P حتما برقرار است، چون آن را می-دانیم و حتما رخ داده است. اما عکس آن صادق نیست: ممکن است P صادق باشد، اما ما آن را ندانیم.

نیز می توان این منطق را منطق زمان دانست و P را چنان خواند که "از این پس P". دیده می شود که این ادعا ادعای کلی تری است از P. چون P می گوید که "در این لحظه P برقرار است" که حالت خاصی از حکم p است. پس از این رو اصل موضوعی چون P →P در منطق موجهات اعتبار عمومی ندارد. ممکن است کسی بگوید که P می تواند معنای خاص تری از Pداشته باشد. چنین نیست. حتی اگر P را این گونه تعریف کنیم که "درست در همین لحظه من دیدم که P". واقعیت این است که P هم چیزی بیش از این نمی گوید. مثلا از ادعای P نمی توانیم بفهمیم که در لحظه ای دیگر یا از دیدگاهی دیگر آیا هنوز این ادعا برقرار است یا نه. اگر چنین اصل موضوعی را بپذیریم، پذیرفته‌ایم که اگر چیزی در این لحظه درست باشد، همیشه درست است، یا پذیرفته ایم که اگر چیزی رخ دهد، حتما ما آن جا بوده ایم و آن را دیده ایم. در این صورت ما مفهوم زمان یا مکان را که می خواهیم در چنین منطق جدیدی حضور داشته باشد، از بین برده ایم. منطق های کلاسیک چنین ویژگی ای ندارند. از زمان و مکان عاری اند. حال که ما می خواهیم این قید و شرط ها را وارد کنیم، پس نباید P →P را بپذیریم.

اما از سوی دیگر اگر ما در یک دستگاه استنتاج منطق موجهات P را استنتاج کرده باشیم. در این صورت ما خواهیم توانست P را نیز استنتاج کنیم، اما نه لزوما در همان لحظه از زمان. یا اینکه آن را خواهیم دید، اما نه لزوما در همان جهان (یعنی P →P برقرار نیست). این نشان می دهد که مراحل

استنتاج در منطق موجهات خود مانند گذر زمان به دانش ما در باره ی آنچه که در یک دستگاه منطق موجهات می توانیم دانست، می‌افزاید.

از همین جا در می یابیم که قضیه‌ی استنتاج در منطق‌های موجهات نمی‌تواند و نباید که درست باشد. چون در غیر این صورت از $P \vdash P$ که قانون استنتاج در منطق موجهات بوده و صادق است باید به دست آید $\vdash P \rightarrow P$ که گفتیم چنین نیست.

با چشم‌پوشی از این پس زمینه‌ی فلسفی منطق های موجهات، زبان کار آمد و در عین حال تصمیم پذیری برای مدل های محاسبه‌ی ترتیبی و موازی به دست داده‌اند. عملگرهای آنها را می‌توان به عنوان عملگرهایی روی جبر بول تعبیر کرد. به این منظور معناشناسی‌های نسبی مانند مدل کریپکی[23] ساختارهای نسبی را به کار می‌برد مانند جهان‌های ممکن، لحظات زمان، وضعیت‌های رایانه.

منطق موجهات استدلال صوری درباره ی کارکرد برنامه‌ها و توصیف حالات گذار بین وضعیت های آن را امکان پذیرتر می کند، از این روست که در علوم رایانه کاربرد زیادی یافته؛ امروزه منطق های موجهات کاربرد گسترده ای در هوش مصنوعی، امنیت اطلاعات، پروتکل های ارتباطی، پردازش زبان های طبیعی پیدا کرده اند. از سوی دیگر منطق‌های شهودگرایانه به خاطر ماهیت ساختنی[24] خود، راه را به علوم رایانه بیش از سایر منطق ها باز کرده‌اند. به همین نحو می توان انتظار داشت که *الگوریتم های استدلال خودکار*[25] برای ترکیب این دو بهتر کار کنند.

در این فصل به معرفی منطق های موجهات گزاره ای (مرتبه صفر) و محمولی (مرتبه ۱) خواهیم پرداخت [7].

---

[23] Kripke semantics

[24] Constructive

[25] Automated reasoning

۱،۱. نحو در منطق موجهات

زبان منطق موجهات عبارت است از مجموعه ای از نمادهای گزاره ای مانند t_1، t_2، و ... ، و تعدادی نمادهای عملگر :

الفبای زبان منطق موجهات: مجموعه‌ای از نمادها است که به چهار دسته افراز می‌شود:

- *گزاره‌های پایه*[۲۶] یا اتمی: A، B، C، ...
- نمادهای صفر موضعی *صدق*[۲۷] و *تناقض*[۲۸]: T و ⊥
- *عملگرها*[۲۹] یا *رابطها*[۳۰]: ، ◊، ¬، ∨, ∧، →، ↔ و ...
- *پرانتزها*[۳۱] یا *نشان گذارها*[۳۲]: (،)، [،] (برای آشنایی بیشتر با این دسته و نکات مربوط به پیاده سازی آنها پیوست ۱ را بخوانید).

در عمل نمادهایی که با آن‌ها کار می‌کنیم متناهی‌اند. پس می‌توانیم همه‌ی آن‌ها را در یک مجموعه‌ی متناهی موسوم به *الفبا*[۳۳]ی زبان منطقی گرد آوریم و آن را با ξ نشان دهیم. از این پس رشته‌ای که قصد ساخت، ارزش دهی یا تجزیه آن را داریم، تنها از این نمادها ساخته شده‌است، یعنی رشته‌ای است از ξ^*.

$$\xi = \{A ،B ،C... ،) ،(،] ، [، \quad ،\Diamond ،\wedge ,\vee \quad ،\rightarrow ،\leftrightarrow ...\} \qquad (۱ .۱)$$

---

[26] Basic Symbols

[27] Truth

[28] Falsum

[29] Operations

[30] Connectives

[31] Parenthesis

[32] Punctuation

[33] Alphabet

طبعا همه‌ی اعضای ξ^* به درد ما نمی‌خورند. بسیاری از آن‌ها مهمل هستند. تنها آن‌هایی به کار ما می‌آیند که بر اساس قواعد خاصی ترکیب شده باشند. اکنون زبان خاصی به نام $F\xi$ را به نحوی که $F\xi\subset\xi^*$ روی الفبای ξ تعریف می‌کنیم:

بنا بر تعریف، $F\xi$ کوچک ترین مجموعه‌ای است که این سه شرط را داشته باشد:

- شامل دو نماد به نام‌های نماد صفر موضعی *صدق* و تناقض (⊥ و T) باشد. این دو نماد را بعدا معرفی خواهیم کرد.
- شامل نمادهای گزاره‌های اتمی (A، B، C و ...) باشد.
-

ا

گر شامل رشته‌هایی مانند P و Q بود، شامل رشته‌های (P) و (Q) هم باشد. همچنین رشته‌هایی که از ترکیب مستقیم آن‌ها به واسطه‌ی عملگرها پدید می‌آیند، یعنی (¬P)، (P)، (◊P) ، (P∧Q)، (P∨Q)، (P→Q)، (P↔Q) را نیز در بر داشته باشد.

به هر یک از اعضای $F\xi$ یک *فرمول درست‌ساخت*[34] می‌گوییم: فرمول درست‌ساخت یا فرمولی اتمی است، یا نمادی صفر موضعی و یا گزاره‌ای مولکولی که با ترکیب گزاره‌های ساده‌تر توسط رابط-های منطقی ساخته شده‌است.

۲،۱. معنا در منطق موجهات

[34] Well Formed Formula *(wff)*

مدل معنایی که در اینجا مورد بحث است مدلی دو ارزشی است [11] . فیتینگ در [9] روی خانواده ای از منطق های موجهات مرتبه ی صفر چند ارزشی و روش های استنتاج خودکار آنها بحث مفصلی کرده است.

تفاوت تعریف عملگرهای منطقی در منطق موجهات و کلاسیک: عملگرها (تابع‌ها یا رابط‌ها) ی منطقی دو گزاره را با هم ترکیب کرده و یک گزاره‌ی جدید به دست می‌دهند.

در منطق کلاسیک گزاره ای (مرتبه صفر) می توان از *جدول ارزش*[۳۵] برای تعریف متغیرها سود جست. در منطق کلاسیک ارزش نهایی فرمول ساخته شده بر اساس عملگرها به عنوان یک گزاره صرفا با دانستن ارزش همه‌ی نمادهای به کاررفته و به یاری جدول ارزش قابل تحقیق است. جدول ارزش یک گزاره‌ی مرکب، جدولی است که در هر سطر آن ارزش آن گزاره بر حسب ارزش گزاره‌های ساده‌ی تشکیل دهنده‌ی آن محاسبه شده باشد. برای گزاره‌ی مرکبی که از چند گزاره‌ی ساده تشکیل شده‌است، هر سطر از جدول ارزش، نشان دهنده‌ی یک تعبیر برای این گزاره‌های ساده است. پس جدول ارزش می‌تواند به عنوان تعریف یک عملگر به کار رود. منطق هایی از این دست را که می توان به صورت تابعی ارزش گزاره ی کلی تر را بر حسب اجزا به دست آورد را منطق های *تابع صدق*[۳۶] می نامند. منطق های موجهات چنین نیستند. خواهیم دید که در یک جهان به خصوص قوانین منطق کلاسیک بر آنها حاکم می شود، اما عملگرهای موجهات کار گذار بین این جهان ها را انجام می دهند. تفاوت اصلی تعریف عملگرها بین این دو منطق مربوط به همین عملگرهای موجهات است که برای گنجاندن آنها در مدل معنایی ناچاریم جهان های مختلفی را تعریف کنیم.

---

[35] Truth table

[36] Truth-Functional

فضای ارزش: در منطق موجهات کلاسیک دو ارزشی (منطق موجهاتی که بر پایه ی منطق کلاسیک دو ارزشی و با افزودن عملگرهای موجهات به دست آید) یک گزاره می‌تواند درست باشد یا نادرست. با چنین تعریفی *مجموعه‌ی ارزش‌دهی*[۳۷] یا *فضای ارزش‌ها*[۳۸] که آن را با Tr نشان می‌دهیم، عبارت از {درست، نادرست} خواهد بود؛ هر گزاره در هنگام ارزش‌دهی به یکی از اعضای این مجموعه‌ی دوعضوی متصل می‌شود.

تعبیر[۳۹]: اگر مجموعه‌ی {... و l={A، B، C مجموعه‌ی نمادهای جمله‌ای اولیه (اتمی) زبان صوری منطق باشد، هر تابع مانند t را که از l در مجموعه‌ی ارزش‌دهی تعریف شود، یک *تعبیر* یا یک *ارزش دهی*[۴۰] از l (مجموعه‌ی نمادهای گزاره‌های اولیه) می‌نامیم.

$$t: \{A, B, C, \dots\} \to Tr$$ (۲ .۱)

با تغییر مجموعه‌ی Tr در این تعریف می‌توان مجموعه‌ی ارزش دهی را با غیر از دو عضو تشکیل داد. ما می‌توانیم مثلا منطق سه ارزشی داشته باشیم.

دستگاه معنایی یک منطق موجهات شامل عملگرهای   و ◊ عبارت است از:

- مجموعه‌ای از جهان ها $W = \{w_1, w_2, \dots\}$
- یک رابطه‌ی دسترسی به صورت $R \subseteq W\times W$
- برای هر دنیای w یک تابع تعبیر (ارزش‌دهی) مانند $t_w: \{A, B, C, \dots\} \to Tr$ که چندان که پیشتر هم گفتیم Tr مجموعه ی ارزش های ما است که در اینجا آن را {0,1} در نظر می گیریم.

[37] Set of validity

[38] Pase of values

[39] interpretation

[40] Valuation

قاب[41]: به زوج مرتب $F=(W, R)$ قاب گفته می شود. در واقع مجموعه ی جهان‌هایی که داریم و مجموعه ی روابط دسترسی بین آنها را قاب می نامیم.

مدل[42]: زوج مرتب $M=(F, t)$ یا سه تایی مرتب $M= (W, R, t)$ را یک مدل می گویند . مدل در واقع همان قاب است که تابع تعبیر (ارزش دهی) t را هم با خود دارد. این تابع برای هر یک از جملات اتمی در هر یک از جهان های w ارزش تعیین می کند.

ارزش‌دهی به یک فرمول درست‌ساخت: اینک نحوه ی ارزیابی فرمول های درست ساخت بر این مبنا به این صورت خواهد بود:

هر فرمولی از زبان منطق موجهات در *جهان فعلی*[43] اگر در همان مرحله ی اول تجزیه فاقد عملگر های و ◊ باشد (عملگر فعال آن در این مرحله از تجزیه هیچ یک از این دو عملگر نباشد)، با همان روال منطق گزاره ای ارزیابی می شود:

- $t_w(P)=1$ اگر و تنها اگر $t(P)$ در دنیای w درست باشد.
- $t_w(\neg P)=1$ اگر و تنها اگر $t(\neg P)$ در دنیای w درست باشد.
- $t_w(P\wedge Q)=1$ اگر و تنها اگر $t(P\wedge Q)$ در دنیای w درست باشد.
- $t_w(P\vee Q)=1$ اگر و تنها اگر $t(P\vee Q)$ در دنیای w درست باشد.
- $t_w(P\rightarrow Q)=1$ اگر و تنها اگر $t(P\rightarrow Q)$ در دنیای w درست باشد.

...

41 Frame

42 Model

43 Current World

بنابر این، عملگرهای منطقی گزاره ای به همان نحو گذشته تعبیر می شوند. الان هنگام تعبیر عملگرهای   و ◊ فرا رسیده است.

برای هر $w\in W$:

$$t_w(\ P) = \begin{cases} 1 & ;\ \ \forall w': (w,w')\in R \rightarrow \\ 0\ ; & \text{otherwise} \end{cases}$$

$$t_w(\Diamond P) = \begin{cases} 1 & ;\ \ \exists w': (w,w')\in R \wedge t \\ 0\ ; & \text{otherwise} \end{cases}$$

از این تعاریف می‌توانیم رابطه ی بین عملگرهای   و ◊ را نیز استخراج کنیم. برای نمونه:

$t_w(\neg\ \ P)=1$

$\Leftrightarrow\ \neg\ (t_w(\ \ P)=1)$ (یا $t_w(\ \ P)=0$)

$\Leftrightarrow\ \neg\forall w': (w,w')\in R \rightarrow t_{w'}(P)=1$

$\Leftrightarrow\ \ \exists w': (w,w')\in R \wedge t_{w'}(P)=0$

$\Leftrightarrow\ \exists w': (w,w')\in R \wedge t_{w'}(\neg P)=1$

$\Leftrightarrow\ t_w(\Diamond\neg P)=1$

این به معنای هم ارزی $\neg\ \ P\equiv\Diamond\neg P$ است. به همین ترتیب نیز می‌توان هم‌ارزی $\Diamond P\equiv\neg\ \ \neg P$ را هم به دست آورد.

تعبیر یک فرمول درست ساخت در این زبان بر اساس تعبیر گزاره‌های اتمی به کار رفته در آن و با استفاده از تابع t که بالاتر تعریف کردیم انجام می شود. برای نمونه:

$t_w(\neg\ \ (P\vee Q))=1$

$$\Leftrightarrow \quad \neg(\forall w': (w,w')\in R \rightarrow t_{w'}(P\vee Q)=1)$$
$$\Leftrightarrow \quad \exists w': (w,w')\in R \wedge t_{w'}(P\vee Q)=0$$
$$\Leftrightarrow \quad \exists w': (w,w')\in R \wedge (t_{w'}(P)=0 \vee t_{w'}(P)=0)$$
$$\Leftrightarrow \quad \exists w': (w,w')\in R \wedge (t_{w'}(P)=0 \vee t_{w'}(P)=0)$$

یعنی $P\vee Q$ در جهان w لزوما درست نیست، اگر جهانی چون w' وجود داشته که از w قابل دستیابی بوده و در آن P یا Q نادرست باشند.

در واقع در هر جهان در ارزشیابی فرمول‌هایی که عملگر فعال آنها عملگر غیر موجهات (کلاسیک) باشد، منطق حاکم منطق کلاسیک است:

- $M,w \models P$ اگر و تنها اگر، در جهان w داشته باشیم $t \models P$
- $M,w \models P\wedge Q$ اگر و تنها اگر $M,w \models P$ و $M,w \models Q$
- $M,w \models P\vee Q$ اگر و تنها اگر $M,w \models P$ یا $M,w \models Q$
- $M,w \models \neg P$ اگر و تنها اگر $M,w \not\models P$
- $M,w \models P\rightarrow Q$ اگر و تنها اگر $M,w \not\models P$ یا $M,w \models Q$

اینک درجات اعتبار یک فرمول را در منطق های موجهات می کنیم:

ارضا شدن[44]: فرمول P را در جهان w از مدل $M=(W, R, t)$ ارضا شده می نامیم اگر در جهان $w\in W$ تحت تعبیر داده شده ی t درست باشد: $M, w \models P$

درست[45]: فرمول P را در $M=(W, R, t)$ درست نامیم اگر در هر $w\in W$ درست باشد: $M \models P$

معتبر[46]: فرمول P را معتبر نامیم اگر در تمام مدل ها درست باشد: $\models P$

ارضاپذیر[47]: فرمول P را ارضا پذیر گوییم، اگر مدلی مانند $M=(W, R, t)$ وجود داشته باشد که در جهانی مانند w از آن ارضا شده باشد. (یا به عبارت دیگر $\neg P$ معتبر نباشد).

---

44 Satisfied

45 True

46 Valid

47 Satisfiable

ارزش‌دهی به مجموعه‌ای از فرمول‌های درست‌ساخت: فرض کنیم Σ مجموعه‌ای از فرمول-های درست‌ساخت از زبان منطق گزاره‌ها باشد ($\Sigma \subseteq F\xi$) و $M= (W, R, t)$ باشد به‌صورتی که در t برای همه‌ی فرمول‌های اتمی تشکیل‌دهنده‌ی اعضای Σ باشد. در این صورت ارزش این مجموعه از فرمول‌ها در مدل M درست است اگر همه‌ی اعضای آن تحت این تعبیر درست باشند.

۳،۱. دستگاه های اصل موضوعی در منطق موجهات

در فصل دوم به تعریف مفصل دستگاه‌های اصل موضوعی هیلبرتی خواهیم پرداخت. در این بخش تنها به تعریفی مختصر در این باره و بیان های مختلف اما هم ارزی که برای منطق کلاسیک گزاره ای شده است می پردازیم. یک دستگاه اصل موضوعی برای یک منطق عبارت است از اصول موضوعه[۴۸] و قوانینی[۴۹] به این شرح که اصول موضوعه مجموعه ای از فرمول های درست ساخت هستند که به طور بدیهی در منطق مورد نظر درست فرض می شوند. مجموعه‌ای از قوانین استنتاج قوانینی هستند که روی اصول موضوعه اعمال می شوند و قضایای جدیدی را استنتاج می‌کنند. مجموعه ی این قضایای استنتاج شده را نظریه[۵۰] می نامیم. به این ترتیب هر بیانِ اصل موضوعی از یک منطق می تواند تمام قضایای معتبر در آن منطق را به دست دهد. در واقع تعریف عملگرها در این روش با استفاده از خود اصول موضوعه انجام می‌شوند و این اصول هستند که معنای عملگرها را در خود نگه داری می کنند. از این رو است که برای تعریف هر منطق می توان صرفا بیان اصل موضوعی آن را به کار برد. الان می خواهیم این کار را برای منطق کلاسیک گزاره ای انجام دهیم. بیان های مختلفی از این منطق موجود اند که همه هم ارز اند و بیانگر همان دستگاه PL هستند.

---

[48] Axioms

[49] Rules

[50] Theory

قوانین استنتاج معتبر:

جایگزینی یکنواخت: بر اساس این قانون حاصل جایگزینی هر فرمول درست ساخت در یک قضیه به جای یک فرمول درست ساخت دیگر خود یک قضیه است. به عبارت دیگر اگر به جای هر تکرار یک فرمول درست ساخت (یک زیررشته) از یک قضیه (یک رشته) یک فرمول درست ساخت دیگر را قرار دهیم، حاصل باز هم یک قضیه است و در منطق مورد بحث معتبر می باشد. در واقع این امکان وجود دارد که قانون جایگزینی را تنها برای اصول موضوعه اعمال کنیم و از نظر فنی این قانون جایگزینی محدودشده تفاوتی در نتیجه نخواهد داشت، اما هزینه ی محاسباتی را ممکن است افزایش دهد. در پیاده سازی روش استنتاج اصل موضوعه ای که در فصل دوم به آن خواهیم پرداخت ، قانون جایگزینی محدود تر را به کار برده ایم. این قانون را US می نامیم.

Modus Ponens : اگر P و P→Q قضایایی از دستگاه باشند، Q هم هست. در واقع عملگر → تنها عملگری است که معنای اش را از قانون MP می گیرد. قوانین این دستگاه در مورد هیچ عملگر دیگری اظهار نظر نمی کنند و آن را به اصول موضوعه (در واقع تعدادی رشته) وا می گذارند. دستگاه هایی که در ادامه خواهند آمد، هر یک بیانی از منطق گزاره‌ای کلاسیک اند. نرم افزار ما می‌تواند بر مبنای همه‌ی آنها با استفاده از الگوریتم استنتاج قضایا در دستگاه های هیلبرتی کار کند. برای خوانایی بیشتر برای هر اصل موضوعه یک نام مشخص گذاشته ایم که نه تنها درون همان سیستم که بیرون از آن نیز با همان نام شناخته شود. X، Y، Z و ... می توانند با هر فرمول درست‌ساختی جایگزین شوند و لزوما اتمی نیستند (اتمی‌ها را با A، B، C و ... نشان دادیم).

پس از بیان اصول موضوعه قوانین آن دستگاه را ذکر می‌کنیم که در اینجا فقط یک قانون MP است. قانون US را در تعریف دستگاه های هیلبرتی گنجانده‌ایم و دیگر نیازی به ذکر آن از این پس

نخواهد بود. قانون MP هم چندان که پیدا است دو گونه بیان می‌شود: یا بر مبنای عملگر استنتاج ($X,X\rightarrow Y|Y$) و یا برمبنای عملگر عطف. این امر بسته به آن است که اصول مان را بر مبنای کدام یک از اینها بنویسیم. نکته ای که در اینجا مهم است این است که اصول باید برمبنای عملگرهایی نوشته شوند که تمامیت تابعی[51] دارند. مثلا $\{\neg,\vee\}$ تمامیت تابعی دارد. برای تعریف عملگرهای دیگر دو راه وجود دارد. یا اصولی برای آنها و به طور جداگانه بنویسیم، برای نمونه H4 و H5 در دستگاه های ۸ و ۹ از فهرست زیر عملگر $\wedge$ را تعریف می‌کنند. یک راه دیگر تعریف عملگر است بر مبنای عملگرهایی که در اصول موضوعه تعریف شده‌اند. در این روش باید در فرمول اولیه‌ای که قصد بررسی اثبات آن را داریم، هر عملگری که در اصول بازتاب نداشته را بر مبنای عملگرهای تعریف شده بازنویسی کنیم و وقتی که فرمول ما فقط شامل عملگرهای تعریف شده بود، تنها با همان اصل ها کار کنیم. برای نمونه به جای آن دو اصول H4 و H5 می توانیم مثل دستگاه ۱ از جایگزینی $X\wedge Y$ is $\neg(\neg X\vee\neg Y)$ استفاده کنیم.

اصول موضوعه‌ی منطق کلاسیک:

به دستگاه‌های هم ارز زیر توجه کنید. همه ی اینها با هم معادل اند و در واقع منطق گزاره ای کلاسیک را نشان می دهند که منطق موجهات بر پیایه آنها ساخته می شود. در فصل ۵ نشان خواهیم داد که چگونه می توان منطق موجهاتی را بر اساس منطق شهودگرایانه استوار کرد:

1. SKC ($\neg,\vee$)

$[S]\neg X\vee(Y\vee X)$

$[K]\neg(\neg X\vee(\neg Y\vee Z))\vee(\neg(\neg X\vee Y)\vee(\neg X\vee Z))$

$[C]\ \neg X\vee X$

$[MP]X,\neg X\vee Y|Y$

[51] Functionally Completeness

0 is $\neg(\neg\neg\neg\ X \vee X)$

1 is $(\neg\neg\neg\ X \vee X)$

$X \wedge Y$ is $\neg(\neg X \vee \neg Y)$

$X \to Y$ is $(\neg X \vee Y)$

$X \equiv Y$ is $\neg(\neg(\neg X \vee Y) \vee \neg(\neg Y \vee X))$

2. SKC ($\neg$,$\to$)

[S]$X \to (Y \to X)$

[K]$(X \to (Y \to Z)) \to ((X \to Y) \to (X \to Z))$

[C]$\neg\neg\ X \to X$

[MP]$X, X \to Y | Y$

0 is $\neg(X \to (X \to X))$

1 is $(X \to (X \to X))$

$X \vee Y$ is $(\neg X \to Y)$

$X \wedge Y$ is $\neg(X \to \neg Y)$

$X \equiv Y$ is $\neg((X \to Y) \to \neg(Y \to X))$

3. SKC (0,$\to$)

[S]$X \to (Y \to X)$

[K]$(X \to (Y \to Z)) \to ((X \to Y) \to (X \to Z))$

[C]$((X \to 0) \to 0) \to X$

[0]$0 \to X$

[MP]$X, X \to Y | Y$

1 is $(0 \to 0)$

$\neg X$ is $(X \to 0)$

$X \vee Y$ is $((X \to 0) \to Y)$

$X \wedge Y$ is $((X \to (Y \to 0)) \to 0)$

$X \equiv Y$ is $(((X \to Y) \to ((Y \to X) \to 0)) \to 0)$

4. Principia Methematica ($\vee, \to, 0$)

[R1]$(X \vee X) \to X$

[R2]$Y \to (X \vee Y)$

[R3]$(X \vee Y) \to (Y \vee X)$

[R4]$(Y \to Z) \to ((X \vee Y) \to (X \vee Z))$

[0]$0 \to X$

[MP]$X, X \to Y | Y$

1 is $(0 \to 0)$

$\neg X$ is $(X \to 0)$

$X \wedge Y$ is $((X \to 0) \vee (Y \to 0)) \to 0)$

$X \equiv Y$ is $((X \to Y) \wedge (Y \to X))$

5. Lukasiewicz ($\neg, \to$)

[CN1]$(X \to Y) \to ((Y \to Z) \to (X \to Z))$

[CN2]$X \to (\neg X \to Y)$

[CN3]$(\neg X \to X) \to X$

[MP]$X, X \to Y | Y$

0 is $\neg (X \to (\neg X \to X))$

1 is $(X \to (\neg X \to X))$

$X \wedge Y$ is $\neg (X \to \neg Y)$

$X \vee Y$ is $(\neg X \to Y)$

$X \equiv Y$ is $((X \to Y) \wedge (Y \to X))$

6. Rosser ($\neg,\wedge,\rightarrow$)

[KN1]$X\rightarrow(X\wedge X)$

[KN2]$(X\wedge Y)\rightarrow X$

[KN3]$(X\rightarrow Y)\rightarrow(\neg(Y\wedge Z)\rightarrow\neg(X\wedge Z))$

[MP]$X,X\rightarrow Y|Y$

0 is $\neg(X\rightarrow(X\wedge X))$

1 is $(X\rightarrow(X\wedge X))$

$X\vee Y$ is $(\neg X\rightarrow Y)$

$X\equiv Y$ is $((X\rightarrow Y)\wedge(Y\rightarrow X))$

7. Andrews ($\neg,\vee$)

[P1]$\neg(X\vee X)\vee X$

[P2]$\neg X\vee(Y\vee X)$

[P3]$\neg(\neg X\vee Y)\vee(\neg(Z\vee X)\vee(Y\vee Z))$

[MP]$X,\neg X\vee Y|Y$

0 is $\neg(\neg(X\vee X)\vee X)$

1 is $(\neg(X\vee X)\vee X)$

$X\wedge Y$ is $\neg(\neg X\vee\neg Y)$

$X\rightarrow Y$ is $(\neg X\vee Y)$

$X\equiv Y$ is $\neg(\neg(\neg X\vee Y)\vee\neg(\neg Y\vee X))$

8. Semifull Classical ($0,1,\rightarrow,\vee,\wedge$)

[H1]$X\rightarrow(Y\rightarrow X)$

[H2]$(X\rightarrow(Y\rightarrow Z))\rightarrow((X\rightarrow Y)\rightarrow(X\rightarrow Z))$

[H3]$X\rightarrow(Y\rightarrow(X\wedge Y))$

[H4]$(X\wedge Y)\rightarrow X$

[H5] $(X \wedge Y) \rightarrow Y$

[H6] $X \rightarrow (X \vee Y)$

[H7] $Y \rightarrow (X \vee Y)$

[H8] $(X \rightarrow Z) \rightarrow ((Y \rightarrow Z) \rightarrow (X \vee Y) \rightarrow Z)$

[H9] $0 \rightarrow X$

[H10] $X \rightarrow 1$

[H11] $((X \rightarrow 0) \rightarrow 0) \rightarrow X$

[MP] $X, X \rightarrow Y | Y$

$\neg X$ is $(X \rightarrow 0)$

9. Full Classical $(0,1,\neg,\rightarrow,\vee,\wedge)$

[H1] $X \rightarrow (Y \rightarrow X)$

[H2] $(X \rightarrow (Y \rightarrow Z)) \rightarrow ((X \rightarrow Y) \rightarrow (X \rightarrow Z))$

[H3] $X \rightarrow (Y \rightarrow (X \wedge Y))$

[H4] $(X \wedge Y) \rightarrow X$

[H5] $(X \wedge Y) \rightarrow Y$

[H6] $X \rightarrow (X \vee Y)$

[H7] $Y \rightarrow (X \vee Y)$

[H8] $(X \rightarrow Z) \rightarrow ((Y \rightarrow Z) \rightarrow (X \vee Y) \rightarrow Z)$

[H9] $0 \rightarrow X$

[H10] $X \rightarrow 1$

[H11] $\neg\neg X \rightarrow X$

[H12] $(X \rightarrow 0) \rightarrow \neg X$

[H13] $\neg X \rightarrow (X \rightarrow 0)$

[MP] $X, X \rightarrow Y | Y$

اصول موضوعه ی منطق موجهات:

هنگام آن رسیده که اصل های مربوط به عملگرهای موجهات را به دستگاه های یادشده بیافزاییم (چون منطق های موجهات تعمیم‌هایی از منطق کلاسیک هستند که عملگرهای جدیدی را وارد کار می کنند، آنچه را که راجع به عملگرهای پیشین می گویند باید صادق باشد. در اینجا اصول موضوعه را به نام هایی بیان می کنیم و سپس می گوییم افزودن کدام اصول به منطق کلاسیک کدام یک از منطق های موجهات را می سازد. این به آن معنا است که منطق های موجهاتی که ارائه خواهیم کرد لزوما هم ارز نیستند. اصول زیر را در نظر بگیرید:

[K1] $(X\to Y)\to(\ X\to\ Y)$

[K2] $(X\wedge Y)\leftrightarrow(\ X\wedge\ Y)$

[K3] $(\ X\vee\ Y)\to\ (X\vee y)$

به منطقی که علاوه بر اصول و قوانین منطق کلاسیک (هر یک از آن نه دستگاه ۳،۱،۱.) اصول فوق را دارا باشد، منطق موجهات K می گوییم. این منطق ضعیف ترین منطق موجهات نرمال است. همه ی منطق های موجهات نرمال این سه اصل را دارا هستند و برخی اصول دیگری را نیز به خود اضافه می کنند. منطق های موجهات غیرنرمالی هم وجود دارند که در آنها حتی در این سه اصل هم تشکیک می شود. این منطق ها در نهایت می گویند "هیچ چیز لازم نیست" و "هر چیزی ممکن است". بحث ما در اینجا روی منطق های موجهات نرمال می چرخد. در هر صورتی برای وارد کردن عملگر ◊ باید آن را بر مبنای   تعریف کنیم:

$\Diamond X$ is $\neg\ \ \neg X$

این کار را می توانیم همچنین با افزودن اصل موضوع زیر انجام دهیم:

[K4] $\Diamond X\leftrightarrow\neg\ \ \neg X$

پنج اصل مهم دیگری که با افزوده شدن برخی از آنها به K منطق‌های موجهات دیگری پدید می-آیند عبارت اند از:

[M] $X \to X$

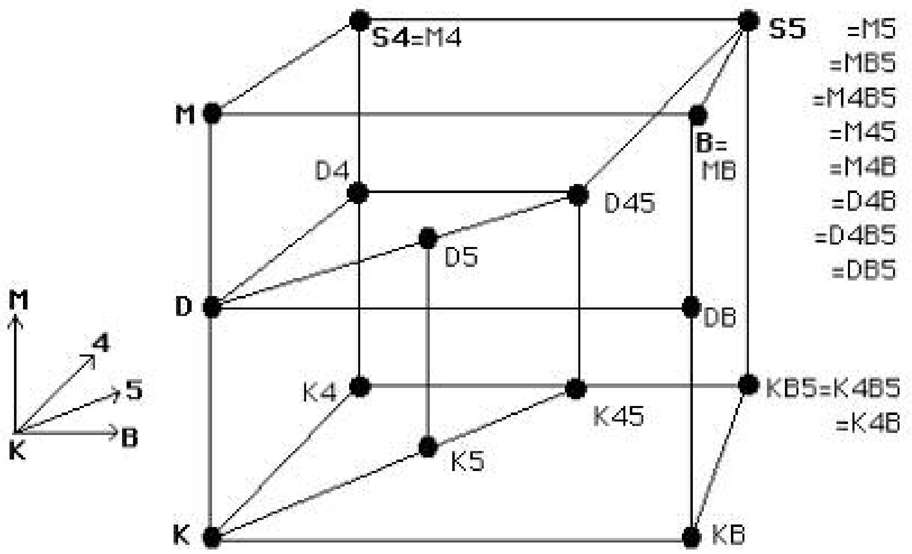
[4] $X \to \quad X$

[5] $\Diamond X \to \ \Diamond X$

[B] $X \to \ \Diamond X$

[D] $X \to \Diamond X$

شکل زیر گویای این است که چگونه می توان منطق‌های مختلفی را با افزودن همین اصول به K سامان داد. هر نقطه منطق موجهات متمایزی از دیگری است. دیده می‌شود که مثلا دستگاه S5 را به چندین صورت مختلف از ترکیب این اصول می توان ساخت.



شکل ۱،۱. نمودار تشکیل برخی از منطق های موجهات بر مبنای اصول موضوعه ی M، D، 4، 5 و B

در [16] روشی برای مقایسه ی دو منطق موجهات داده شده به صورت دستگاه اصل موضوعی ارائه شده است.

۲

دستگاه استنتاج اصل موضوعی هیلبرتی

در این فصل ابتدا به تعریف دستگاه های هیلبرتی می پردازیم. در مورد تعریف کامل سیستم‌های استنتاج هیلبرتی به بخش 2.3.1 از [10]ارجاع می دهیم. از آنجایی که معرفی اولیه ی منطق ها بر مبنای دستگاه هیلبرتی سنتی پذیرفته شده است. در این فصل برای نشان دادن اینکه چگونه دستگاه هیلبرتی را به صورت یک روش استنتاج رو به جلو استفاده کرده‌ایم، ساده ترین تعریف منطق کلاسیک مرتبه ی صفر (گزاره ای) را بررسی و تحلیل خواهیم کرد. در پیوست ۴ می توانید نمونه هایی از پیاده سازی این بخش را مشاهده کنید.

۱،۲. دستگاه استنتاج اصل موضوعی هیلبرت چیست؟

در هر منطق مورد بحث، *دستگاه استنتاج اصل موضوعی*[۵۲] زوج مرتبی مانند (A, R) است که در آن A مجموعه‌ای از رشته‌ها (فرمول‌های درست ساخت آن منطق) موسوم به *اصول موضوع*[۵۳] بوده و

---

[52] Hilbert-Style

[53] Axioms

R هم مجموعه‌ی قواعد استنتاج است که هر یک از اعضای آن رابطه‌ای است بین مجموعه‌ای از رشته‌ها (به عنوان مقدمات یک قاعده) و یک فرمول (به عنوان مفروضات یک قاعده) [10] .

یک استنتاج از هر مجموعه‌ی *مفروضات*[۵۴] موسوم به Γ ، دنباله‌ای ناتهی مانند $\varphi_۱$، $\varphi_۲$، ... و φ_n است به گونه‌ای که برای هر i بین ۱ تا n رشته‌ی φ_i در یکی از این سه شرط بگنجد:

- ب

  ه مجموعه‌ی اصول موضوعه تعلق داشته باشد: $\varphi_i \in A$

- ب

  ه مجموعه‌ی مفروضات متعلق باشد: $\varphi_i \in \Gamma$

- ا

  ز فرمول‌های پیشین با یکی از قواعد استنتاج به دست آمده باشد:

$$(\exists r \in R)(\exists \Delta \subseteq \{ \varphi_1, \varphi_2, \ldots, \varphi_{i-1} \})\Delta r \varphi_i \qquad (۲. ۱)$$

در این صورت گوییم $\Gamma \vdash_{(A,R)} \varphi_n$ یعنی فرمول φ_n در دستگاه (A, R) به طریقه‌ی صوری از مجموعه‌ی مفروضات Γ استنتاج می‌شود. در صورتی که تنها با یک دستگاه اصل موضوعی سر و کار داشته باشیم، به اختصار نگارش $\Gamma \vdash \varphi_n$ را به کار می‌بریم. همچنین اگر $\Gamma=\varnothing$، گوییم فرمول φ_n یک قضیه از (A, R) است و می‌نویسیم $\vdash_{(A,R)} \varphi_n$ یا به اختصار $\vdash \varphi_n$ به این ترتیب می‌توان در یک رویه-ی کامپیوتری، با مشخص نمودن زوج مرتب (A, R) قضایای دستگاه را یک به یک تولید کنیم

اعضای مجموعه ی A در واقع شما[۵۵] هستند. یعنی به جای متغیرهای آنها می توان هر فرمول درست ساختی را گنجاند. در واقع عملا به جای مجموعه‌ی A نه تنها همان اصول موضوعه‌ی متناهی بلکه تعداد بی‌شمار جایگزینی های آن را هم در نظر می گیریم.

---

[54] Hypotheses

خواهیم گفت که مجموعه‌ی قضایای یک دستگاه اصل موضوعی، مجموعه‌ای *بازگشتی شمارش پذیر یا شمارا*[56] است. این به آن معنا است که قضایا فهرست پذیر هستند و هر قضیه را با صبر کافی می‌توان به دست آورد. اما در حالت کلی این مجموعه *بازگشتی*[57] نیست. یعنی همواره نمی‌توان تحقیق نمود که یک فرمول داده شده به مجموعه‌ی قضایا متعلق هست یا نه. فقط می‌دانیم که اگر باشد، با صبر کافی می‌توانیم آن را به دست آوریم و مادامی که آن را به دست نیاورده‌ایم، نمی‌دانیم که یک قضیه هست و هنوز به آن نرسیده‌ایم و یا آنکه نیست.

۲،۲. نکات فنی در پیاده سازی دستگاه استنتاج اصل موضوعی

در اینجا به روشی اشاره می‌کنیم که در پیاده سازی دستگاه های اصل موضوعی به کار گرفته ایم.

جایگزینی بخش مهمی از پیاده سازی الگوریتم است و باید با هشیاری صورت بگیرد. برای این بخش دو ماشین مختلف را باید پیاده سازی کنیم: ماشین فرمول ساز و ماشین جایگزین:

ماشین فرمول ساز: این در واقع ماشینی است که فرمول های درست ساخت را در زبان داده شده یک به یک تولید می کند. هیچ فرمولی را جا نمی اندازد و هیچ فرمولی را هم دو بار تولید نمی کند. زبان داده شده مجاز است عملگرهای صفر، یک و دوتایی داشته باشد. ورودی این ماشین متغیرهایی است که فرمول ما بر اساس آن باید ساخته شوند (مثلا دو متغیر A و B). الگوریتم های متفاوتی را می‌توان برای این ماشین پیشنهاد داد. همه‌ی این الگوریتم‌ها کمابیش فرمول‌های درست-ساخت متشکل از نمادهای مورد کاربرد ما را به اعداد طبیعی یکتایی نسبت می دهند، چندان که

---

[55] Scheme

[56] Recursively Enumerable

[57] Recursive

هیچ دو فرمولی به یک عدد منتسب نشوند (عکس آن لازم نیست). سپس اعداد را به ترتیب شمرده و فرمول مربوطه را بر اساس عدد به دست آمده استخراج می کنند.

عدد گودل خود نمونه ی خوبی است. گودل برای متناظر کردن اعداد طبیعی به فرمول ها از این روش استفاده کرد: وی ابتدا به هر نمادی از زبان که با آن سر و کار داریم یک عدد طبیعی منحصر به فرد نسبت داد. سپس رشته ی $x_1x_2x_3...x_n$ را با عدد $p_1^{x_1}p_2^{x_2}p_3^{x_3}...p_n^{x_n}$ نشان داد. با تجزیه‌ی هر عدد طبیعی به عوامل اول اش می‌توان فرمول اصلی را بازتولید کرد که البته این روش فعلا دو اشکال دارد: اول آنکه ممکن است x_i تعریف نشده داشته باشد. دوم آنکه لزوما فرمول های درست ساخت را به دست نمی دهد. یعنی پس از به دست آمدن هر فرمول باید درست ساخت بودن آن را بررسی کنیم و اگر پاسخ مثبت بود فرمول را به خروجی بفرستیم.

روش دیگر اصلاح شده‌ی روش فوق است: تنها سه عدد $p_1=2$، $p_2=3$ و $p_3=5$ را در نظر می گیریم. اعداد طبیعی را یک به یک تجزیه می کنیم. اگر تنها مرکب از ۲، ۳ و ۵ بودند آنها را به صورت $2^{x_1}3^{x_2}5^{x_3}$ تجزیه می کنیم. x_2 باید عدد اولی باشد که عملگر ما را مشخص کرده و x_1 و x_3 هم باز دو فرمول هستند که باید به طریق بازگشتی تجزیه شوند تا به اعداد منحصر به فردی برسند که برای ثابت ها (عملگرهای صفرتایی) یا گزاره های اتمی در نظر گرفته بودیم.

روش های ترکیبی و مؤثر تری هم موجود اند که برای پیاده سازی کارآمدتر ناچاریم آنها را به کار بگیریم چون روش های فوق بسیار کند عمل می کنند و اکثر قریب به اتفاق اعداد طبیعی در آنها به فرمول های درست ساخت متناظر نمی شوند.

در اینجا با چشم پوشی از اینکه هزینه ی محاسبه برای شمارش فرمول ها چقدر است، تتنها ذکر این نکته کافی است که این مسأله قابل محاسبه با ماشین تورینگ است. تعریف این ماشین و کلاس توابع قابل محاسبه با آن را در [3]، [4]، [12] و [13] بخوانید. بحثی که باقی می‌ماند این است که

چگونه برای اصل استقرا فرمول‌ها را شمارش کنیم. این خود نیاز به زیروال دیگری دارد که هزینه‌ی پیاده‌سازی آن با تورینگ بسیار هنگفت است. در همین پروژه با استفاده از زبان *دلفی*[58] که از تورینگ اکیدا قوی‌تر نیست این کار در یک برنامه‌ی طولانی انجام شده که ترجمه‌ی آن به کد تورینگ نامعقول است. در اینجا تنها به این نکته اشاره می‌کنیم که شمارش فرمول‌ها تورینگ محاسبه‌پذیر است. بدون اینکه ماشین تورینگ آن را معرفی کنیم: مجموعه‌ی فرمول‌های درست ساخت مرکب از نمادهای Σ مجموعه‌ای مستقل از متن است. از سوی دیگر همه‌ی رشته‌های مرکب از نمادهای Σ هم منظم اند. پس می‌توان زیر روالی ساخت که همه‌ی فرمول ها را تولید کرده و تنها آنها که درست ساخت هستند را فهرست کند.

ماشین جایگزین: این ماشین شماهای اصل موضوعی اولیه را را دریافت می‌کند و تمامی جایگزینی‌های ممکن آن را یک به یک به دست می‌دهد. در واقع ماشین جایگزین باید ماشین فرمول ساز را در دل خود داشته باشد. اما از آنجا که ماشین فرمول ساز متوقف نمی‌شود، باید این پردازش به طریق موازی انجام شود. یعنی به ازای تعداد مشخصی از فرمول های تولید شده، تمامی ترکیب‌های آنها در شمای مورد نظر (بسته به اینکه از چند حرف تشکیل شده است) جایگزین شوند. سپس تعدادی فرمول دیگر ساخته شود. و باز این بار همه‌ی جایگزینی های ممکن روی کل مجموعه فرمول های به دست آمده تا کنون، منهای جایگزینی‌هایی که در دور قبل به دست آمده اند تولید شوند. و به همین ترتیب.

به این نحو با ترکیب دو ماشین فوق اعضای مجموعه‌ی A (تمامی جایگزینی های ممکن اصل موضوع های داده شده) به دست خواهد آمد.

---

[58] Delphi

می دانیم که همه‌ی اعضای مجموعه‌ی A قضایای سیستم هستند. اما منطقی که تنها همین قضایا را داشته باشد به هیچ کار نمی‌آید. مفیدترین قضایای ما آنهایی هستند که از روی همین قضایای به دست آمده و با قانون های استنتاج به دست می آیند.

می توان قوانین استنتاج را هم به صورت یک رشته به نرم‌افزار مورد طراحی وارد کرد. در واقع نرم افزار ما طوری طراحی شده است که قانون MP را به صورت رشته‌ی [MP]P,P→Q>Q دریافت می‌کند. این نرم افزار می‌داند که منظور ما از P و Q دو فرمول درست ساخت اند. نیز می‌داند که [MP] نام قانون است و نماد < را به عنوان نماد استنتاج در سطح فرازبان[59] می‌شناسد. یعنی می‌داند که از گزاره‌های سمت چپ (و تمام جایگزینی‌های ممکن به جای P و Q) گزاره‌ی سمت راست به دست می‌آید.

برای پیاده‌سازی این قانون در واقع پس از اضافه شدن هر قضیه به مجموعه‌ی قضایای سیستم (چه این قضیه از جایگزینی به دست آمده باشد و چه از یکی از قوانین استنتاج)، باید قضیه‌ی جدید را با یک یک قضایای قبلی در یک قیاس در نظر بگیریم: آیا این دو می‌توانند به نحوی به صورت P و P→Q نوشته شوند؟ یعنی آیا رشته های P و Q ای وجود دارد که این دو به صورت P و P→Q یا P→Q و P نمایش داده شوند؟ اگر پاسخ مثبت بود رشته‌ی Q به عنوان یک قضیه افزوده می‌شود. این کار آن قدر ادامه می‌یابد که قانون MP قابل اعمال کردن نباشد. در این صورت بقیه‌ی قوانین یک به یک اجرا می‌شوند. زمانی که دیگر هیچ قانون استنتاجی‌کار نکرد باید یک جایگزینی دیگر انجام شود و ....

ذکر این نکته لازم است که پیاده‌سازی برخی از قوانین استنتاج هم باید به طریق موازی صورت گیرد. قانون Nec می گوید: [Nec]X> X. اگر به این قانون اولویت دهیم، رشته‌های X،X ، X و

[59] Meta-Language

... تا ابد تولید می‌شوند و هرگز به اعمال قوانین یا اصل‌های دیگر نخواهیم رسید. در این موارد هم باید از پردازش موازی استفاده کرد. برای هر قضیه‌ای که به هر نحوی اضافه می‌شود باید بلافاصله یک بار این قانون را اجرا کنیم. اما قضیه‌ی به دست آمده در این مرحله را جا بیاندازیم و بعد یک گام دیگر طی نماییم و دوباره بازگردیم که همین قانون را برای فرمول جا انداخته شده اجرا کنیم. این کار فرآیند استنتاج را کند می‌کند اما تمامیت برنامه را از بین نمی‌برد.

درجه‌ی پیچیدگی فرمول درست‌ساخت: تعداد نمادهای منطقی به کار رفته در هر فرمول درست‌ساخت را *درجه‌ی پیچیدگی*[۶۰] آن فرمول گوییم. با این تعریف می‌توان گفت گزاره‌های اتمی گزاره‌هایی هستند که درجه‌ی پیچیدگی آن‌ها صفر است.

مرتبه‌ی فرمول درست‌ساخت: برای یک فرمول درست‌ساخت مرتبه صفر مانند X تابع *مرتبه*[۶۱] موسوم به $r(\mathrm{x})$ به صورت زیر محاسبه می‌شود:

$$\begin{cases} r(\neg \mathrm{A}) = r(\mathrm{A}) = \text{۰} \quad (\text{که A یک گزاره‌ی اتمی به جز } \bot \text{ و } \top \text{ است.}) \\ r(\top) = r(\bot) = \text{۰} \\ r(\neg \top) = r(\neg \bot) = \text{۱} \\ r(\neg\neg \mathrm{Z}) = r(\mathrm{Z}) + \text{۱} \\ r(\mathrm{A} \wedge \mathrm{B}) = r(\mathrm{A}) + r(\mathrm{B}) + \text{۱} \\ r(\mathrm{A} \vee \mathrm{B}) = r(\mathrm{B}) + r(\mathrm{B}) + \text{۱} \end{cases} \qquad (\text{۱ .۱})$$

[60] Complexity Degree

[61] Rank

درختِ نحو: می‌توان مراحل ترکیب فرمول‌های درست‌ساخت کوچکتر را به منظور ساختن فرمول‌های بزرگتر را به صورت یک نمودار درختی موسوم به درخت *نحو*[۶۲] نمایش داد. این کار دقیقا مطابق با تعریف فرمول درست‌ساخت انجام می‌شود. ساده ترین درخت‌های نحو به عملگر (رابط)های منطقی مربوط اند. این رابط‌ها به تعداد موضع (ورودی)های شان فرزند دارند. نماد هر رابط برچسب گره‌ی مربوط به آن رابط است و نام ورودی‌های آن برچسب فرزندان.

⊥ A

اتم‌ها و رابط‌های ۰ تایی برگ اند. (⊥) **رابط یک موضعی**

↔

A B

رابط دودویی، دو فرزند دارد. (**A↔B**)

شکل۱.۲ . درخت‌های نحو ابتدایی

در منطق کلاسیک بر اساس ارزش فرمول‌های اتمی به کار رفته در فرمول درست‌ساخت، می‌توان ارزش یک فرمول درست‌ساخت را تعیین نمود. هماهنگ با تعریف استقرایی فرمول‌های درست‌ساخت، الگوریتم بازگشتی‌ای وجود دارد که ارزش یک فرمول درست‌ساخت را محاسبه کند. چنین الگوریتمی

[62] Syntax Tree

از برگ‌های درخت نحو آغاز کرده و ارزش هر گره را به عنوان برچسب آن گره در نظر می‌گیرد. سپس بر اساس تعریف هر عملگر، ارزش گره‌های بالاتر را بر حسب گره‌های پایین‌تر محاسبه می‌کند تا به ریشه برسد. برچسب ارزش ریشه همان ارزش عبارت خواهد بود. این الگوریتم در صورتی به درستی کار می‌کند که ارزش همه‌ی برگ‌ها (گزاره‌های اولیه) داده شده باشد. این‌ها همان اعضای مجموعه‌ی l={A، B، C و ...} هستند.

چنین کاری برای منطق موجهات امکان پذیر نیست. دلیل آن را جلوتر خواهیم گفت. اما درخت های نحو را کماکان نیاز داریم چون آنها را برای تجزیه ی فرمول های درست ساخت و بررسی هر یک در یک جهان مشخص و تشخیص دامنه های اعمال به کار می بریم، هر چند که منطق ما Truth-Functional نباشد.

دامنه‌ی عمل: یک عملگر روی دامنه‌های عمل خود اعمال می‌شود:

اگر تابع منطقی یک موضعی (عملگر یکانی) ~ در فرمول درست‌ساختی مانند P دیده شود، حتما فرمول درست‌ساختی مانند Q وجود دارد به قسمی که Q~ بحشی از P است. به Q *دامنه‌ی عمل* عملگر ~ گویند. همچنین اگر تابع دودویی * در فرمول درست‌ساختی مانند P وجود داشته باشد، حتما فرمول‌های درست‌ساختی مانند Q_1 و Q_2 وجود دارند به گونه ای که (Q_1*Q_2) بحشی از P است. Q_1 و Q_2 هم *دامنه‌ی عمل* عملگر * نامیده می‌شوند.

یکی از مشکلات ما در پیاده سازی روال های اثبات خودکار پیدا کردن دامنه ی عمل عملگر ها و نتیجتا ساختن درخت نحو است. این کار به پشتیبانی قضیه‌ی یگانه خوانی در منطق مرتبه‌ی صفر بدون ابهام و به صورت یکتا انجام می شود[۶۳]: هر فرمول درست‌ساخت X یا گزاره‌ی اتمی است، یا به صورت $(\neg P)$ است که P یک فرمول درست‌ساخت یکتا است و یا دقیقا به یکی از صورت‌های $(P\wedge Q)$،

[63] Unique Parsing Theorem

(P∨Q)، (P→Q)، (Q←P)، (P↔Q) است، به طوری که P و Q فرمول‌های درست‌ساخت یکتا هستند. (پیوست ۲)

درخت استنتاج[۶۴]: با پیاده سازی گرامر تولید فرمول‌های درست‌ساخت منطق مرتبه صفر، می-بینیم که این گرامر از نوع مستقل از متن است و بنابراین هر رشته از زبان آن (هر فرمول درست-ساخت) یک *درخت استنتاج* دارد که مراحل به دست آمدن آن رشته را بر اساس نماد آغازین (ریشه-ی درخت) و قوانین استنتاج (مجموعه‌ی یال‌های منشعب از یک رأس) نشان می‌دهد. در چنین درختی رأس‌هایی که برگ اند پایانه هستند و رئوس دیگر غیرپایانه. این درخت با درخت نحو که در اینجا با آن سر و کار داریم متفاوت است. درخت نحو تعداد رئوس کمتری دارد، چون همه‌ی رئوس آن پایانه هستند. برگهای درخت نحو یا رابطهای ۰ تایی اند و یا نمادهای گزاره‌ای پایه یا گزاره‌های اتمی (A، B، C و ...). رأس‌های دیگر رابطهای یک موضعی به بالا هستند. به هر حال اگر برای یک فرمول نتوانیم هر یک از این دو نوع درخت را به دست دهیم، آن فرمول درست‌ساخت نیست و همین را می توان در پیاده سازی نرم افزار به عنوان آزمونی برای درست ساخت بودن فرمول وارد شده پیش از آزمودن اثبات پذیری آن به کار بست.

تجزیه ی نحوی فرمول های ساخته شده در گرامر:

می خواهیم گرامری بسازیم که همه‌ی فرمول‌های درست‌ساخت منطق مرتبه‌ی صفر را به دست دهد. چون اعضای الفبای ξ (پایانه‌ها) خود شامل حروف بزرگ و رابطهای منطقی مانند → می‌شوند، پایانه‌ها را با خودشان (اعضای ξ) با **قلم سیاه** و غیرپایانه‌ها را به صورت عبارتهای با *قلم ایتالیک* داخل <> نشان می‌دهیم. همچنین اشتقاق را در قوانین اصلی گرامر با دو نقطه (:) نشان می‌دهیم. با توجه به تعریف فرمول درست‌ساخت منطق گزاره‌ها، قوانین اصلی گرامر خواسته شده عبارتند از:

[64] Parse tree

1. <*basic proposition*>: **A** | **B** | **C** | ...
2. <*basic proposition*>: ⊥ | T
3. <*binary connective*>: ∧ | ∨ | → | ← | ↔
4. <*unary connective*>: ¬ | | ◊
5. <*wff*>: <*basic proposition*>
6. <*wff*>: **(**<*basic proposition*>**)**
7. <*wff*>: **(**<*unary connective*> <*wff*>**)**
8. <*wff*>: **(**<*wff*> <*binary connective*> <*wff*>**)**

این گرامر نشان می‌دهد که زبان منطق موجهات مرتبه صفر، یک زبان *مستقل از متن معین* است. یعنی ابهام ندارد و می‌توان هر رشته از این زبان، یعنی هر فرمول درست‌ساخت را به صورت یکتا تجزیه نمود و مراحل استنتاج آن را نشان داد. از همین امر می توان برای تجزیه ی صوری فرمول هایی استفاده کرد که نرم افزار ما قصد دارد صحت آنها را نشان دهد. برای این کار باید درخت نحو را بر اساس فرمول داده شده بسازیم (پیوست ۳).

۳،۲. سیستم SKC به عنوان نمونه ای گزاره ای برای پیاده سازی:

این سیستم را در فصل گذشته تعریف کردیم. سیستم هیلبرتی SKC از چهار اصل موضوعه‌ی زیر تشکیل شده است:

[S] X→(Y→X)

[K] (X→(Y→Z))→((X→Y)→(X→Z))

[C] ((X→⊥)→⊥)→X

[⊥] ⊥→X

تنها قانون استنتاج در این سیستم، MP است که از دو رشته‌ی x و x→y در دنباله‌ی اثبات، رشته‌ی y را نتیجه می گیرد. این سیستم معادل صحیح و تمام منطق گزاره‌ای کلاسیک دوارزشی است. گفتنی است که دو عملگر → (دوتایی) و ⊥ (صفرتایی) در منطق گزاره‌ای کلاسیک دوارزشی تمامیت تابعی دارند و همه‌ی توابع را با آنها می‌توان ساخت. اصل چهارم برای پوشاندن معنای رابط صفرموضعی ⊥افزوده شده و ضروری است.

۴،۲. تحقیق استقلال اصول دستگاه هیلبرتی

در اینجا به اختصار روشی معرفی می‌کنیم که نشان می دهد آیا صول موضوعه‌ی یک منطق از هم مستقل‌اند یا خیر. یعنی آیا هیچ یک از آنها با استفاده از قوانین استنتاج و دیگر اصول به دست نمی‌آیند. هر کس بخواهد منطقی جدید را به این روش معرفی کند، بهتر است که این کار را برای منطق‌اش انجام دهد تا از زیاده گویی در معرفی اصول خودداری کرده باشد و تنها اصل های لازم را معرفی کند. این کار در حالت کلی و برای هر دستگاه هیبلرتی یک روش کلی دارد که منتسب به پُست[۶۵] است. برای این کار کافی است یک مدل معنایی ارائه دهیم که:

۱- اصل موضوع کنارگذاشته شده در این مدل معنایی همانگویی نباشد. یعنی اصل موضوعی که می‌خواهیم استقلال آن را از اصول دیگر نشان دهیم، در این مدل مثال نقض داشته باشد. مثلا اگر مدل معنایی ما تابع درستی است، دست کم یک سطر از جدول ارزش آن اصول (یک انتساب به اتم-هایش) وجود داشته باشد که آن را نقض کند (مقداری به جز مقدار قراردادی همانگویی‌ساز را به آن نشان دهد) و اگر مدل معنایی ارائه شده تابع درستی نیست، بسته به مدل مثال نقضی داشته باشد که نشان دهد این اصل در این مدل معتبر نیست.

[65] Post

۲- سه اصل دیگر در این مدل همانگویی باشند؛ مدل معنایی ما هر سه اصل دیگر را تأیید کند.

۳- قانون های استنتاج همانگویی بودن را حفظ کند. یعنی نشان داده شود که مثلا در مورد قانون MP ، اگر دو رشته‌ی x و x→y در این مدل معنایی همانگویی باشند، حتما رشته‌ی y هم در آن همانگویی است.

شرط‌های ۲ و ۳ نشان می دهند که سیستم هیلبرتی مبتنی بر سه اصل باقی مانده و قانون MP با توجه به معنای ارائه شده صحیح[66] است، چرا که هر *فرمول درست‌ساختی که در این سیستم استنتاج شود، در مدل معنایی یادشده هم همانگویی خواهد بود.* حال چون بر اساس شرط ۱ اصل موضوع کنارگذاشته شده در این مدل همانگویی نیست، به برهان خلف نمی‌تواند در این سیستم استنتاج پذیر باشد و از سه اصل دیگر نتیجه شود.

باید توجه داشت که *تمامیت*[67] سیستم ضروری نیست، مگر در مواردی که مدل معنایی پیشنهاد شده تابع درستی نبوده و تحقیق شرط سوم (حفظ همانگویی توسط MP) نیاز به استفاده از تمامیت داشته باشد. در صورت یافتن مدل معنایی مناسب با ویژگی‌های یادشده، روش پیشنهادی را می‌توان برای هر سیستم هیلبرتی با هر تعداد اصل و قانون تعمیم داد.

---

[66] Sound

[67] Completeness

۳

روش تابلو برای منطق های موجهات

در این بخش ما روش تابلو[68]ی معنایی را برای منطق های موجهات معرفی می کنیم [1].

تعریف: برای هر فرمول A فرمول های FA و TA را فرمول های علامت دار می نامیم.

تعریف: فرض کنیم A یک فرمول و $M= (W, \leq, R, \models)$ یک مدل باشد. برای هر جهان w که $w\in W$ به کار می بریم:

i) $w \models TA,$ if $w \models A$

ii) $w \models FA,$ if $\neg (w \models A)$

iii) $w \models \delta_1 A, \delta_2 B$ if $w \models \delta_1 A$ and $w \models \delta_2 B$

iv) $w \models \delta_1 A | \delta_2 B$ if $w \models \delta_1 A$ and $w \models \delta_2 B$

[68] Tableau

تعریف: یک فرمول علامت دار[69] را *ارضاپذیر با* I^* [70]گوییم، به نحوی که یک مدل I^* چون $(W, \leq, R, \models)$ و یک w از اعضای W وجود داشته باشد به گونه ای که $w \models A$.

مجموعه‌ی Ω از فرمول‌های علامت دار را *ارضاپذیر با* I^* می نامیم، اگر یک مدل I^* و یک جهان w عضو W وجود داشته باشد، چندان که برای هر A عضو Ω داشته باشیم: $w \models A$.

مجموعه‌ی Ω از فرمول‌های علامت دار را *ارضاناپذیر با* I^* می نامیم، اگر ارضاپذیر با I^* نباشد.

قوانین ساختاری[71]:

$$[\text{w-rule}]\ \frac{\Gamma}{\Delta}\ \text{with}\ \Gamma \supseteq \Delta \tag{۳. ۱}$$

$$[\text{Cut}]\ \frac{\Gamma}{\Gamma, TA \mid \Gamma, FA}\ \text{with}\ \Gamma \supseteq \Delta \tag{۳. ۲}$$

قوانین منظم[72]:

$$[\text{R}\wedge]\ \frac{\Gamma, TA \wedge B}{\Gamma, TA, TB} \qquad \frac{\Gamma, FA \wedge B}{\Gamma, FA \mid \Gamma, FB} \tag{۳. ۳}$$

$$[\text{R}\vee]\ \frac{\Gamma, TA \vee B}{\Gamma, TA \mid \Gamma, TB} \qquad \frac{\Gamma, FA \vee B}{\Gamma, FA, FB} \tag{۳. ۴}$$

$$[\text{R}\neg]\ \frac{\Gamma, T\neg A}{\Gamma, FA} \tag{۳. ۵}$$

$$[\text{R}\to]\ \frac{\Gamma, TA \to B}{\Gamma, FA \mid \Gamma, TB} \tag{۳. ۶}$$

---

69 Signed

70 I^* Satisfiable

71 Structural

72 Regular

قوانین موجهات:

$$[X]\ \frac{\Pi}{\Omega_1\ |\Omega_2|\ldots|\ \Omega_n}$$

تعریف: فرض کنیم Γ مجموعه ای از فرمول‌های علامت دار باشد، در این صورت:

$\Gamma^{\#}=\{TA : A\in\Gamma\}$ (۷ .۳)

نیز دو مجموعه ی زیر را تعریف می‌کنیم:

$T\Diamond\Gamma=\{T\Diamond A\ |\ TA\in\Gamma\}$ (۸ .۳)

$F\ \ \Gamma=\{F\ \ A\ |\ FA\in\Gamma\}$ (۹ .۳)

حال فرض کنیم Ψ مجموعه ای از فرمول‌های بی علامت باشد. مجموعه های زیر را تعریف می‌کنیم:

$\Psi^{T}=T\Psi=\{TA\ |\ A\in\Psi\}$ (۱۰ .۳)

$\Psi^{\perp}=F\Psi=\{FA\ |\ A\in\Psi\}$ (۱۱ .۳)

$T\Diamond\Psi=\{T\Diamond A\ |\ TA\in\Psi\}$ (۱۲ .۳)

$F\Diamond\Psi=\{F\Diamond A\ |\ FA\in\Psi\}$ (۱۳ .۳)

$T\ \ \Psi=\{T\ \ A\ |\ TA\in\Psi\}$ (۱۴ .۳)

$F\ \ \Psi=\{F\ \ A\ |\ FA\in\Psi\}$ (۱۵ .۳)

$\Gamma^{N}=$ (۳. ۲)

for K, D, T as	$\{TA : T\Box A \in \Gamma\} \cup \{FA : F\Diamond A \in \Gamma\};$
for K4, KD4 as	$\{TA, T\Box A : T\Box A \in \Gamma\} \cup \{FA, F\Diamond A : F\Diamond A \in \Gamma\};$
for KB, KDB, B as	$\{TA : T\Box A \in \Gamma\} \cup \{FA : F\Diamond A \in \Gamma\} \cup$ $\cup \{T\Diamond A : TA \in \Gamma\} \cup \{F\Box A : FA \in \Gamma\};$
for S4 as	$\{T\Box A : T\Box A \in \Gamma\} \cup \{F\Diamond A : F\Diamond A \in \Gamma\};$
for KB4 as	$\{TA, T\Box A : T\Box A \in \Gamma\} \cup \{FA, F\Diamond A : F\Diamond A \in \Gamma\} \cup$ $\cup \{T\Diamond A : TA \in \Gamma\} \cup \{F\Box A : FA \in \Gamma\} \cup$ $\cup \{T\Diamond A : T\Diamond A \in \Gamma\} \cup \{F\Box A : F\Box A \in \Gamma\};$
for K5, KD5 as	$\{TA : T\Box A \in \Gamma\} \cup \{FA : F\Diamond A \in \Gamma\} \cup$ $\cup \{T\Diamond A : T\Diamond A \in \Gamma\} \cup \{F\Box A : F\Box A \in \Gamma\};$
for K45, KD45 as	$\{TA, T\Box A : T\Box A \in \Gamma\} \cup \{FA, F\Diamond A : F\Diamond A \in \Gamma\} \cup$ $\cup \{T\Diamond A : T\Diamond A \in \Gamma\} \cup \{F\Box A : F\Box A \in \Gamma\};$
for S5 as	$\{T\Box A : T\Box A \in \Gamma\} \cup \{F\Diamond A : F\Diamond A \in \Gamma\} \cup$ $\cup \{T\Diamond A : T\Diamond A \in \Gamma\} \cup \{F\Box A : F\Box A \in \Gamma\}.$

قوانین π

[S◊] $$\frac{\Gamma, T\Diamond A}{\Gamma^{N}, TA}$$ (۳. ۱۶)

[S] $$\frac{\Gamma, F\ \ A}{\Gamma^{\#}, T\Diamond\Delta_{1}, F\ \ \Omega_{1} \mid \ldots \mid \Gamma^{\#}, T\Diamond\Delta_{p}, F\ \ \Omega p \mid \Gamma^{\#N}, T\Delta_{p+1}, F\Omega_{p+1} \mid \ldots \mid \Gamma^{\#N}, T\Delta_{n}, F\Omega_{n}}$$

(۳. ۱۷)

که در

$$\frac{\Gamma^{\#N}, FA}{\Gamma^{\#N}, T\Delta_{1}, F\Omega_{1} \mid \ldots \mid \Gamma^{\#N}, T\Delta_{n}, F\Omega_{n}}$$ (۳. ۱۸)

قوانین υ

[R] $$\frac{\Gamma, T\ \ A}{\Gamma, TA}$$ (۳. ۱۹)

[R◊] $$\frac{\Gamma, F\Diamond A}{\Gamma, FA}$$ (۳. ۲۰)

[Sυ] $$\frac{\Gamma}{\Gamma^{N}}$$ (۳. ۲۱)

زنجیره: زنجیره[73] ای از Γ به دنباله ی متناهی ای از فرمول های علامت دار گفته می شود، چندان که $\Gamma_0=\Gamma$ بوده و برای هر $0\leq i\leq n$ قانونی وجود داشته باشد به طوری که Γ_i پایین خط و Γ_{i+1} بالای خط قرار گیرد.

گره: به هر حلقه از زنجیره یک گره گوییم.

گره‌ی بسته: به یک گره بسته[74] گوییم، اگر A وجود داشته باشد که شامل TA و FA هر دو باشد.

یک I^*T-تابلو برای یک Γ مجموعه ی τ از زنجیره های Γ است به گونه‌ای که برای هر زنجیره‌ی t که $t\in\tau$ و برای هر گره‌ی Γ_i که $\Gamma_i\in t$ و $i>0$، مجموعه‌ی فرمول‌های پایین خط مربوط به قانونی که به Γ_{i-1} اعمال شده است $(\Gamma_{i-1}\in t)$، گره های زنجیره ای چون $t'\in\tau$ باشد.

تابلوی بسته: یک تابلو بسته است، اگر هر زنجیره‌ای یک گره‌ی بسته داشته باشد. یک تابلو بسته برای Γ یک اثبات برای Γ است.

تعریف: یک I^*-T-تابلو یک اثبات I^*-T برای فرمول بی علامتی چون X است، اگر اثباتی برای {FX} باشد.

برای نمونه می‌خواهیم $\neg\Diamond A\to\neg A$ را ثابت کنیم:

$F\neg\Diamond A\to\ \neg A$

$[S\to]$

$T\neg\Diamond A, F\neg\ A$

$[S]$

---

[73] Thread

[74] Closed

۱،۳. صحت و تمامیت روش تابلو

قضیه (صحت روش تابلو): فرض کنیم Γ مجموعه ای از فرمول های علامت دار باشد. اگر Γ یک اثبات T-I^*-تابلو داشته باشد در آن صورت Γ ارضاناپذیر با I^* است. (در اینجا * معرف سیستم های K، D، T، KB، KDB، B، K4، KD4، S4، KB4، K5، KD5، K45، KD45 و S5 می باشد.)

اثبات: می دانیم که یک تابلو ارضاپذیر است، اگر برای بعضی از زنجیره هایش، همه ی گره ها ارضا پذیر باشند. در واقع باید ثابت کنیم که اگر T ارضاپذیر باشد، در آن صورت یک تابلو مانند T' به دست آمده از T با اعمال یکی از قوانین تابلو نیز ارضاپذیر است. در مورد صحت قوانین ساختاری و منظم در بیشتر کتاب های مقدماتی استدلال خودکار از جمله [10] یا [7] بحث شده است. در [8] صحت قوانین موجهات و شهودی مورد بررسی قرار گرفته است.

برای اثبات صحت قوانین موجهات:

$S\Diamond$) فرض کنیم که Γ و $T\Diamond A$ ارضاپذیر باشد. فرض کنیم w جهانی است که که $w|=\Gamma,F\ \ A$ در این صورت جهانی چون $w^{\bullet}$ وجود دارد به گونه ای که $w^{\bullet}|=TA$. می‌دانیم که $w^{\bullet}|=\Gamma^N$ و بنابراین Γ^N و TA ارضاپذیر است.

S) p را انتخاب می کنیم چنان که $p\leq n$ باشد. فرض کنیم جهان w به گونه ای که $w|=\Gamma,F\ \ A$ وجود داشته باشد. در این صورت برای جهانی چون $w^{*\bullet*}$ و یک i که $i\leq n$ داریم $w^{*\bullet*}|=\Gamma^{\#N},T\Delta_I,$ $F\Omega_i$. در این صورت بنا بر فرض ما $p>I$ است. مادامی که $w^*Rw^{*\bullet}\leq w^{*\bullet*}$ ، جهانی چون u وجود دارد به گونه ای که $w^*\leq uRw^{*\bullet*}$ بنابراین $w\leq u$ و در نتیجه $u|=\Gamma^{\#}$. از سوی دیگر داریم $u|=T\Diamond\Delta_i,F\ \ \Omega_i$ که تناقض است و حکم ثابت می شود.

برای اثبات تمامیت به [1] مراجعه کنید.

۴

روش محاسبه رشته‌ها برای منطق های موجهات

چنان که گفتیم ،دستگاه‌های هیلبرتی اشکالات فراوانی دارند. در بخش پایانی این پایان نامه آزمون‌هایی از مقایسه‌ی روش‌های گوناگون برای فرمول‌های یکسان به دست داده شده است. در مورد برخی از این فرمول‌ها چندان که خواهیم دید این روش در مدت زمان معقول تعیین شده به نتیجه نرسیده اند.

در این بخش به معرفی یک فرمول بندی به سبک گنتزن[۷۵] برای منطق‌های موجهات می‌پردازیم. وی مبدع سیستم‌های رشته ای[۷۶] بود و از این ایده استفاده کرد که در روش‌هایی مانند دستگاه اصل موضوعی هیلبرتی و یا استنتاج طبیعی[۷۷] جستجو برای اثبات یک فرمول بدون آگاهی از این که این فرمول به لحاظ معنایی معتبر[۷۸] است یا خیر انجام می پذیرد. اگر ما به روشی غیر مستقیم و بدون

---

[75] Gerhard Gentzen

[76] Sequent system

[77] Natural Deduction

[78] Valid

ساختن اثبات بدانیم که فرمولی معتبر است، از این آگاهی می توانیم برای ساختن اثبات استفاده کنیم.

تعریف: برای دو مجموعه‌ی فرمول‌های درست‌ساخت Γ و Δ به عبارت $\Gamma\rightarrow\Delta$ یک رشته[۷۹] گوییم. Γ را مقدم[۸۰] و Δ را تالی[۸۱] می‌نامیم. هر فرمول در یکی از این دو مجموعه یک فرمول رشته ای[۸۲] است. $\Gamma\rightarrow\Delta$ ادعا می‌کند که از مجموعه‌ای از همه‌ی فرمول‌های Γ برخی از فرمول‌های Δ به دست می آید.

اصل موضوع:

$$\Gamma, A \rightarrow A, \Delta \qquad (۴. ۱)$$

قوانین ساختاری[۸۳]:

تضعیف[۸۴]

$$[\mathrm{Lw}]\ \frac{\Gamma \rightarrow \Delta}{D, \Gamma \rightarrow \Delta} \qquad [\mathrm{Rw}]\ \frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, D}$$

انقباض[۸۵]

$$[\mathrm{Lc}]\ \frac{D, D, \Gamma \rightarrow \Delta}{D, \Gamma \rightarrow \Delta} \qquad [\mathrm{Rc}]\ \frac{\Gamma \rightarrow \Delta, A, A}{\Gamma \rightarrow \Delta, A}$$

جابجایی[۸۶]

[79] Sequent

[80] Antecedents

[81] Succedents

[82] Sequent formula

[83] Structural rules

[84] Weakening

[85] Contraction

$$[\mathrm{Le}]\ \frac{\Gamma, C, D, \Gamma \to \Delta}{\Gamma, D, C, \Gamma \to \Delta} \qquad [\mathrm{Re}]\ \frac{\Gamma \to \Delta, C, D,}{\Gamma \to \Delta, D, C,}$$

برش[87]

$$[\mathrm{Cut}]\ \frac{\Gamma \to \Delta, A - A, \Pi \to \Lambda}{\Gamma, \Pi \to \Delta, \Lambda} \qquad (۴.۵)$$

قوانین منطقی[88]:

$$[\mathrm{L}\wedge]\ \frac{C, D, \Gamma \to \Delta}{C \wedge D, \Gamma \to \Delta} \qquad [\mathrm{R}\wedge]\ \frac{\Gamma \to \Delta, A - \Gamma}{\Gamma \to \Delta, A}$$

$$[\mathrm{L}\vee]\ \frac{C, \Gamma \to \Delta - D, \Gamma \to \Delta}{C \vee D, \Gamma \to \Delta} \qquad [\mathrm{R}\vee]\ \frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \vee B}$$

$$[\mathrm{L}\neg]\ \frac{\Gamma \to \Delta, A}{\Gamma, \neg A \to \Delta} \qquad [\mathrm{R}\neg]\ \frac{A, \Gamma \to}{\Gamma \to \neg A} \qquad (۸.$$

$$[\mathrm{L}\supset]\ \frac{\Gamma \to \Delta, C - D, \Pi \to \Lambda}{C \supset D, \Gamma, \Pi \to \Delta, \Lambda} \qquad [\mathrm{R}\supset]\ \frac{\Gamma, A \to B}{\Gamma \to A \supset B}$$

قوانین موجهات[89]:

$$[\mathrm{R}\ \ ]\ \frac{\Pi, \Gamma^{v}, \Diamond\Lambda_1 \to \ \Omega_1 \ldots \Pi, \Gamma^{v}, \Diamond\Lambda_p \to \ \Omega_p - \Gamma, \Lambda_{p+1} \to \Omega_{p+1} \ldots \Gamma, \Lambda_n \to \Omega_n}{\Pi, \Gamma^{v} \to \Gamma, \Lambda_{p+1} \to \ A}$$

$(n \geq p \geq 0)$ (۴.۱۰)

$$\frac{\Gamma, \Lambda_1 \to \Omega_1 - \Gamma, \Lambda_n \to \Omega_n}{\ldots} \qquad (۴.۱۱)$$

با اعمال قوانین منطقی و ساختاری

---

[86] Exchange

[87] Cut rule

[88] Logical rules

[89] Modal rules

$$\frac{\ldots}{\Gamma \rightarrow A} \tag{۱۲.۴}$$

و در آن:

$\Gamma^{\nu} = \{\Box A : A \in \Gamma\}$ for K, D, T;
$\Gamma^{\nu} = \{\Box A : A \in \Gamma\} \cup \{\Box A : \Box A \in \Gamma\}$ for KD4, K4;
$\Gamma^{\nu} = \{\Box A : A \in \Gamma\} \cup \{A : \Diamond A \in \Gamma\}$ for KB, KDB, B;
$\Gamma^{\nu} = \{\Box A : \Box A \in \Gamma\}$ for S4;
$\Gamma^{\nu} = \{\Box A : A \in \Gamma\} \cup \{\Box A : \Box A \in \Gamma\} \cup \{A : \Diamond A \in \Gamma\} \cup \{\Diamond A : \Diamond A \in \Gamma\}$
$\Gamma^{\nu} = \{\Box A : A \in \Gamma\} \cup \{\Diamond A : \Diamond A \in \Gamma\}$ for K5, D5;
$\Gamma^{\nu} = \{\Box A : A \in \Gamma\} \cup \{\Box A : \Box A \in \Gamma\} \cup \{\Diamond A : \Diamond A \in \Gamma\}$ for K45, KD45;
$\Gamma^{\nu} = \{\Box A : \Box A \in \Gamma\} \cup \{\Diamond A : \Diamond A \in \Gamma\}$for S5;

$$[L\Diamond]\frac{\Gamma, A \rightarrow \Delta}{\Gamma^{\nu}, \Diamond A \rightarrow \Delta^{\pi}} \tag{۱۳.۴}$$

که در آن:

$\Delta^{\pi} = \{\Diamond A : A \in \Delta\}$ for K, D, T;
$\Delta^{\pi} = \{\Diamond A : A \in \Delta\} \cup \{\Diamond A : \Diamond A \in \Delta\}$ for KD4, K4;
$\Delta^{\pi} = \{\Diamond A : A \in \Delta\} \cup \{A : \Box A \in \Delta\}$ for KB, KDB, B;
$\Delta^{\pi} = \{\Diamond A : \Diamond A \in \Delta\}$ for S4;
$\Delta^{\pi} = \{\Diamond A : A \in \Delta\} \cup \{\Diamond A : \Diamond A \in \Delta\} \cup \{A : \Box A \in \Delta\} \cup \{\Box A : \Box A \in \Delta\}$for KB4;
$\Delta^{\pi} = \{\Diamond A : A \in \Delta\} \cup \{\Box A : \Box A \in \Delta\}$ for K5, D5;
$\Delta^{\pi} = \{\Diamond A : A \in \Delta\} \cup \{\Diamond A : \Diamond A \in \Delta\} \cup \{\Box A : \Box A \in \Delta\}$ for K45, KD45;
$\Delta^{\pi} = \{\Diamond A : \Diamond A \in \Delta\} \cup \{\Box A : \Box A \in \Delta\}$ for S5.

یک اثبات رشته ای I^{*} برای یک رشته ای $\Gamma \rightarrow \Delta$ درختی است از رشته‌ها به گونه ای که:

a) هر رشته از اثبات (هر گره از درخت) رشته ای بالایی از قانونی باشد، به جز یکی که رشته ی پایانی[90] نامیده می شود و همان $\Gamma \rightarrow \Delta$ است.

b) هر رشته از اثبات (هر گره از درخت) رشته ای پایینی از قانونی باشد، به جز آنها که رشته های آغازین[91] نامیده می شود و همان اصول موضوعه هستند.

[90] End sequent

[91] Initial sequents

این درخت از پایین به بالا نمایش داده می شود و اثباتی است برای پایین ترین گره $(\Gamma\rightarrow\Delta)$ که بالاترین گره های آن همان اصل‌ها هستند.

هر اثبات رشته ای I^* یک اثبات رشته ای I^* برای رشته ی $\rightarrow X$ می باشد.

به ترتیب فوق می‌توانیم برای فرمول داده شده‌ی X جهت تحقیق اینکه آیا از قضایای سیستم هست یا $\rightarrow X$ را به عنوان پایین ترین گره در نظر بگیریم. در هر لحظه مشخص است که کدام یک از قوانین اعمال می‌شود، چون با مشخص کردن عملگری که در هر مرحله اولویت دارد و تفکیک دامنه های اعمال آن قانون مربوطه انتخاب می‌گردد. پس از تشکیل کامل درخت، اگر همه‌ی برگ‌ها به فرمت اصل موضوع (۴. ۱) بودند، قضیه به منطق ما تعلق دارد و درخت به دست آمده برهانی برای آن است.

۵

در آمدی بر منطق های موجهات شهودگرایانه

منطق های موجهات شهودگرایانه منطق هایی موجهاتی هستند که منطق زیرین آنها به جای منطق کلاسیک، شهودگرایانه است. برای توضیح آنها نخست باید منطق های شهودگرایانه را تعریف کنیم:

۱،۵. منطق شهودگرایانه:

منطق شهودگرایانه گزاره ای[۹۲] را به اختصار IPL می نامیم. از آنجا که منطق شهودگرایانه با منطق کلاسیک تنها در اصول موضوعه ی دستگاه اثبات اش متفاوت است، نحو در آن مشابه منطق کلاسیک (بند ۱،۱،۱) می باشد. بعدا به تفاوت های دستگاه های اصل موضوعی خواهیم پرداخت. اما معناشناسی این منطق به کلی از منطق کلاسیک متفاوت است. یکی از این معناشناسی ها توسط کریپکی[۹۳] بیان شده است و از معناشناسی ای که وی در مورد منطق های موجهات (که در آینده

[92] Intuitionistic Propositional Logic

[93] Kripke

خواهیم گفت) پیشنهاد کرده بود الهام گرفته شده است. وی معناشناسی اش را بر مبنای تبیین جزییاتی شهودی درباره ی عملگرهای شهودگرایانه استوار کرد. هرچند که اثبات قضیه ی تمامیت نیاز به استدلال کلاسیک دارد، در نتیجه این معناشناسی تنها از دیدگاه کلاسیک قابل فهم است. در اینجا مدلی که وی برای IPL مرتبه صفر معرفی کرد را شرح می دهیم. دستگاه معنایی یک منطق شهودگرایانه عبارت است از:

- مجموعه‌ای از جهان ها $W = \{w_1, w_2, \dots\}$
- یک رابطه‌ی دسترسی ترتیب جزیی[۹۴] به صورت $R \subseteq W \times W$
- برای هر دنیای w یک تابع *ایجاب*[۹۵] مانند $t_w: \{A, B, C, \dots\} \rightarrow Tr$ که Tr مجموعه ی ارزشی ما است (در اینجا تنها شامل ۰ و ۱) و A و B و .. گزاره‌های اتمی هستند. اگر $t_w(A)$ مقدار یک را بگیرد گوییم w گزاره‌ی A را ایجاب می کند[۹۶]. این تابع *سازگار*[۹۷] است به این معنا که وجودندارد w چندان که $t_w(P)=1$ و $t_w(\neg P)=1$ (در یک جهان یک گزاره و نقیض آن توأمان ایجاب نمی‌شوند). نیز این تابع *یکنوا*[۹۸] است به این معنا که اگر $t_w(A)=1$ و $w \leq w'$ در این صورت $t_{w'}(A)=1$. یعنی وقتی یک گزاره در جهانی تأیید شد، در جهان های پس از آن نیز اعتبارش را حفظ می‌کند.

تعریف قاب و مدل مشابه تعریف آنها در مورد منطق موجهات است. یعنی به مجموعه‌ی جهان ها و رابطه ی ترتیب جزیی دسترسی شان قاب و به این قاب به همراه تابع ایجاب در اینجا IPL-مدل[۹۹] می‌گوییم.

94 Partially Ordered

95 Forcing function

96 w forces A

97 Consistent

98 Monotone

99 IPL-model

برای اینکه بدانیم گزاره‌های مرکب (در اینجا P، Q و ...) در جهان های مختلف چگونه ایجاب می-شوند، در منطق مرتبه‌ی صفر قوانین زیر را به کار می گیریم:

- $t_w(P\wedge Q)=1$ اگر $t_w(P)=1$ و $t_w(Q)=1$.
- $t_w(P\vee Q)=1$ اگر $t_w(P)=1$ یا $t_w(Q)=1$.
- $t_w(P\to Q)=1$ اگر برای هر $w'\geq w$ اگر $t_{w'}(P)=1$ آنگاه $t_{w'}(Q)=1$.
- $t_w(\neg P)=1$ اگر وجود نداشته باشد $w'\geq w$ که $t_{w'}(P)=1$.

اکنون، دستگاه‌های نه‌گانه‌ی اصل موضوعی که به عنوان نمونه برای بیان منطق کلاسیک معرفی کردیم را به یاد بیاورید. برخی از آنها اصل موضوع $\neg\neg X\to X$ [C] یا بیان های دیگر آن مانند $X\vee\neg X$ [C] را (موسوم به اصل طرد شق ثالث[100]) مستقیما به عنوان اصل (و نه قضیه) دارا هستند. با حذف این اصل از هر یک از این دستگاه ها، آنچه باقی می ماند دستگاه ضعیف تری است موسوم به منطق شهودگرایانه. طبعا این منطق برخی از قضایای منطق کلاسیک را دربر می گیرد، اما هر چه که این منطق ادعا کند، منطق کلاسیک نیز آن را می پذیرد. این سخت گیرانه تر بودن منطق شهودگرایانه امتیازی برای آن است. چرا که طرفداران این منطق بر مبنای اصول کمتری ریاضیاتی ساختنی استوار کرده اند.

روشن است که چرا اصل C را طرد شق ثالث می نامند. چون این اصل در واقع می گوید اگر چنین نباشد که X نیست، پس X هست؛ حالت سومی هم در کار نیست. منطق شهودگرایانه می گوید که حالت سومی هم هست، زمانی که دستگاه ما نتواند در مورد X یا نقیض آن اظهار نظر کند. این اختلاف منجر می شود که برهان خلف توسط شهودگرایان مورد قبول نباشد.

[100] Excluded middle

در نهایت با حذف این اصل از دستگاه های فوق می توان پنج بیان مختلف برای یک منطق (منطق شهودگرایانه) به صورت اصل موضوعی به دست داد:

1. SK $(\neg,\vee)$

$[S]\neg X\vee(Y\vee X)$

$[K]\neg(\neg X\vee(\neg Y\vee Z))\vee(\neg(\neg X\vee Y)\vee(\neg X\vee Z))$

$[MP]X,\neg X\vee Y|Y$

0 is $\neg(\neg\neg\neg X\vee X)$

1 is $(\neg\neg\neg X\vee X)$

$X\wedge Y$ is $\neg(\neg X\vee\neg Y)$

$X\rightarrow Y$ is $(\neg X\vee Y)$

$X\equiv Y$ is $\neg(\neg(\neg X\vee Y)\vee\neg(\neg Y\vee X))$

2. SK $(\neg,\rightarrow)$

$[S]X\rightarrow(Y\rightarrow X)$

$[K](X\rightarrow(Y\rightarrow Z))\rightarrow((X\rightarrow Y)\rightarrow(X\rightarrow Z))$

$[MP]X,X\rightarrow Y|Y$

0 is $\neg(X\rightarrow(X\rightarrow X))$

1 is $(X\rightarrow(X\rightarrow X))$

$X\vee Y$ is $(\neg X\rightarrow Y)$

$X\wedge Y$ is $\neg(X\rightarrow\neg Y)$

$X\equiv Y$ is $\neg((X\rightarrow Y)\rightarrow\neg(Y\rightarrow X))$

3. SK $(0,\to)$

[S]$X\to(Y\to X)$

[K]$(X\to(Y\to Z))\to((X\to Y)\to(X\to Z))$

[0]$0\to X$

[MP]$X,X\to Y|Y$

1 is $(0\to 0)$

$\neg X$ is $(X\to 0)$

$X\vee Y$ is $((X\to 0)\to Y)$

$X\wedge Y$ is $((X\to(Y\to 0))\to 0)$

$X\equiv Y$ is $(((X\to Y)\to((Y\to X)\to 0))\to 0)$

4. Semifull Intuitionistic $(0,1,\to,\vee,\wedge)$

[H1]$X\to(Y\to X)$

[H2]$(X\to(Y\to Z))\to((X\to Y)\to(X\to Z))$

[H3]$X\to(Y\to(X\wedge Y))$

[H4]$(X\wedge Y)\to X$

[H5]$(X\wedge Y)\to Y$

[H6]$X\to(X\vee Y)$

[H7]$Y\to(X\vee Y)$

[H8]$(X\to Z)\to((Y\to Z)\to(X\vee Y)\to Z)$

[H9]$0\to X$

[H10]$X\to 1$

[MP]$X,X\to Y|Y$

$\neg X$ is $(X\to 0)$

5. Full Intuitionistic $(0,1,\rightarrow,\vee,\wedge)$

[H1]$X\rightarrow(Y\rightarrow X)$

[H2]$(X\rightarrow(Y\rightarrow Z))\rightarrow((X\rightarrow Y)\rightarrow(X\rightarrow Z))$

[H3]$X\rightarrow(Y\rightarrow(X\wedge Y))$

[H4]$(X\wedge Y)\rightarrow X$

[H5]$(X\wedge Y)\rightarrow Y$

[H6]$X\rightarrow(X\vee Y)$

[H7]$Y\rightarrow(X\vee Y)$

[H8]$(X\rightarrow Z)\rightarrow((Y\rightarrow Z)\rightarrow(X\vee Y)\rightarrow Z)$

[H9]$0\rightarrow X$

[H10]$X\rightarrow 1$

[H12]$(X\rightarrow 0)\rightarrow\neg X$

[H13]$\neg X\rightarrow(X\rightarrow 0)$

[MP]$X,X\rightarrow Y|Y$

۲،۵. منطق های موجهات شهودگرایانه

هنوز اجماع درستی بر سر آنکه چه چیزی را منطق موجهات شهودگرایانه بنامیم انجام نشده است. یکی از مشکلات گرد آوری این پیوست هم همین بوده که هر کس منطقی با این نام درست کرده و آن را منتشر کرده که این منطق ها شباهتی در فرم و عملگرهای به کار رفته، اصل‌ها و نیز مدل معنایی با هم ندارند.

منطق های موجهات شهودگرایانه از منابع گوناگون نتیجه شده اند و کاربردهای مختلف یافته اند. برخی موجهات را به منطق شهودگرایانه ، تحت عنوان عملگرهای جدید شهودی افزوده‌اند. برخلاف منطق های موجهات غیر شهودی در منطق های موجهات شهودگرایانه عملگرهای لزوم (◊) و امکان

( ) دوگان هم[۱۰۱] نیستند. این امر امکان های بیشتری برای تعریف منطق های موجهات شهودگرایانه ایجاد می‌کند.

منطق های موجهاتی که در اینجا بحث شدند به عنوان منطق های غیرکلاسیک شناخته می شوند، اما آنها در واقع منطق های کلاسیک هستند. چرا که بر پایه ی منطق های کلاسیک استوار شده اند و تنها عملگرهایی بر آنها اضافه دارند. منطق استدلال آنها چیزی غیر کلاسیک نیست. پس از این پس آنها را منطق های موجهات کلاسیک می نامیم. می توان منطق های موجهات غیر کلاسیک نیز تعریف نمود. در واقع می توان این عملگرها را به منطق شهودگرایانه افزود و چیزی که حاصل خواهد شد یک منطق موجهات شهودگرایانه است. پس این دسته ی جدید منطق ها، منطق های موجهاتی هستند که منطق زیرین[۱۰۲] شان منطق شهودگرایانه است.

پس چندان که منطق موجهات (ML) را برپایه ی PL ساختیم، منطق موجهات شهودگرایانه (IML) را هم به این نحو از منطق شهودگرایانه (IPL) می سازیم:

۱. IML روی IPL محافظه کارانه[۱۰۳] است: محافظه کاری[۱۰۴] روی IPL شرطی است که تنها روی قاب[۱۰۵] های غیر موجهات[۱۰۶] از IML اعمال می شود. هر چند که استدلال شهودگرایانه باید روی کل زبان کار کند.

۲. IML همه‌ی جایگزینی های قضایای IPL را می پذیرد و تحت MP نیز بسته است: بی شک عکس آن برقرار نیست؛ یک قضیه از IML ممکن است به دلایلی موجهاتی توسط IPL استنتاج نشود.

---

101 Dual

102 Underlying

103 Conservative

104 Conservativity

105 Fragment

106 Non modal

۳. اگر اصل طرد شق ثالث یا هم ارز آن را به این منطق بیافزاییم حاصل باید IPL باشد:

این امر منطقی به نظر می رسد. درست کاستن همین اصل از منطق کلاسیک بود که منطق شهودگرایانه را به دست داد. حال اگر ما چنین قدرتی را از یک منطق موجهات معمول بگیریم، حاصل باید یک منطق موجهات شهودگرایانه باشد.

۴. اگر $A \vee B$ قضیه ای از IML باشد، در آن صورت یا A قضیه ای از IML است و یا B:

این از ویژگی های منطق شهودگرایانه بود که در آن به عملگر فصلی معنایی متفاوت از عملگر فصلی کلاسیک می داد. در واقع هنگامی $A \vee B$ در سیستم قابل استنتاج است که یا A استنتاج شده باشد و یا B. در حالی که در منطق کلاسیک برای چنین ادعایی لزوم به اثبات هیچ یک از این دو نیست. چیزی که ترکیب فصلی را تأیید می کند نه محتوای اثباتی اولی است و نه دومی بلکه می تواند چیزی بینابین آن دو باشد که در نهایت از همان اصل طرد شق ثالثی که در اینجا آن را قبول نداریم نشأت گرفته است. خود این اصل نمونه ی مشخصی برای این بند است. ما نه X را به عنوان قضیه می شناسیم و نه $\neg X$ را پس فصل آن دو را نباید بپذیریم. این امر خوانش ساختنی[۱۰۷]تری به "یا" می دهد، چرا که باید دست کم یک طرف آن را بسازیم (اثبات کنیم).

۵. □ و ◊ عملگرهای مستقلی[۱۰۸] در IML هستند: این هم همتای IPL است. چندان که در مورد PL هم انتظار داشتیم مثلا $\vee$ و $\wedge$ مستقل باشند. .

برای دست یافتن به تعریف بهتری از IML این راه حل خوبی است که مشترکات آن را با IPL مشخص کرده و آنهایی را که دلیل قاطعی (از دیدگاه شهودگرایانه) برای باقی ماندن پیدا نمی کنند را حذف کنیم تا ببینیم چه باقی می ماند.

[107] Constructive

[108] Independent

برای نمونه، اینکه قاعده ی الزام از IPL در IML باقی بماند، منطقی به نظر می رسد. به عبارت کلی تر، این را خواهیم پذیرفت که اگر $A\to B$ یک قضیه است، $A\to\ B$ و $\Diamond A\to\Diamond B$ نیز خواهند بود.

در نمونه ای دیگر به طور کلاسیک روی ترکیب عطفی محدود پخش می شود: $T\leftrightarrow T$ و $(A\wedge B)\leftrightarrow(\ A\wedge\ B)$. همچنین $\Diamond$ هم روی ترکیب فصلی محدود پخش می شود: $\Diamond\bot\leftrightarrow\bot$ و $\Diamond(A\vee B)\leftrightarrow(\Diamond A\vee\Diamond B)$. برای اینها هم دلیل شهودگرایانه ی محکمی نداریم که چرا آنها را در IML دور بیاندازیم.

در اینجا به نکته‌ی مهمی در بحرانی بودن تعریف منطق‌های شهودگرایانه (دست کم تا کنون) اشاره می کنیم. روشی که برای تعریف منطق‌های موجهات شهودگرایانه در پیش گرفته ایم، کمابیش مانند تمام روش هایی که دیگران به این منظور به کار گرفته‌اند بر مبنای یک مقایسه[۱۰۹] بین منطق های کلاسیک و موجهات و پیاده‌سازی دوباره‌ی آن بین منطق‌های شهودگرایانه و شهودگرایانه‌ی موجهات بود. اشکالاتی که تا کنون همه به آن برخورده اند به دو گروه زیر تقسیم می شود:

نخست آن که ممکن است بعضی از شرط‌های لازم را تعریف نکرده باشیم. برای مثال هنوز معلوم نیست که همین پیش‌شرط‌هایی که بالاتر ذکر کردیم می توانند رابطه‌ی بین $\Diamond$ و را مانند منطق های موجهات کلاسیک به دست دهند یا خیر.

دوم آنکه ممکن است بعضی از این اصول پذیرفته شده اضافی باشند. یعنی تنها دلیلی که ما آنها را پذیرفتیم (مثل قاعده ی توزیع پذیری و ...) این بود که دلیل شهودگرایانه ی محکمی به ذهن مان نرسیده تا کنون که اینها باید حذف شوند.

مشکل اصلی ، فقدان یک چارچوب همگرا است که در آن مناسب بودن یا نبودن اصول موضوعه ای که نمونه شان را در بالا بحث کردیم، به درستی ارزیابی شوند. چیزی که در اینجا به آن نیاز داریم

---

[109] Analogy

یک معنا شناسی شهودگرایانه ی پیشنهادی برای عملگرهلی موجهات است که با توجه به آن اصول موضوعه‌ی موجهه‌ی گوناگون بتوانند در آن ارزیابی شوند. این خواسته ها را در این پیش شرط عنوان می کنیم:

۶. بیان شهودگرایانه ای از معنای عملگرهای موجهات باید وجود داشته باشد چندان که IML بر مبنای آن صحیح[۱۱۰] و تمام[۱۱۱] باشد.

در اینجا مختصرا به شرح کلی درباره ی کارهای انجام شده و کوشش های انتشار یافته به منظور تدوین منطقی که هم موجهات را پوشش دهد و هم شهودگرایانه باشد خواهیم پرداخت. منبع اصلی تاریخچه [15] می باشد.

نخستین مقاله را در این زمینه فیچ[۱۱۲] در سال ۱۹۴۸ منتشر کرد [6]. وی نسخه ی شهودگرایانه ی مرتبه ی اولی از منطق موجهات T را ارائه کرد. وی هم دستگاه اصل موضوعی هیلبرتی و هم حساب گنتزنی را برای آن اراده داد، اما آنها را با منطق معرفی شده اش تطابق نداد. روش او از جمله روش هایی بود که بر مبنای مقایسه بین منطق های کلاسیک و موجهات استوار شده بود. امروزه ما انتخاب اصل موضوع های او را تا اندازه ای دلخواه تلقی می‌کنیم. وی نه تنها توزیع پذیری ◊ را نپذیرفت، پیش شرط ۵ را هم رد کرد. کار او از نظر پیشتاز بودن ارزشمند است.

دومین پدیدار شدن عبارت منطق موجهات شهودگرایانه به کتاب سال ۱۹۵۷ پرایر[۱۱۳] باز می گردد [14] که همتای شهودگرایانه ی S5 را ساخت و آن را MIPQ نام نهاد. این منطق پیش شرط

---

[110] Sound

[111] Complete

[112] Fitch

[113] Prior

های ۱ تا ۵ را می پذیرفت و با استدلال‌هایی که می‌تواند ۶ را هم ارضا کند. الگوریتم تصمیم گیری MIPQ در مقاله ای از مینک[۱۱۴] در سال ۱۹۶۸ و به زبان روسی ارائه شد.

سومین مقاله از بال[۱۱۵] بود که نسخه ی شهودگرایانه ای از S5 را ارائه داد که نگاهی کلاسیک به موجهات داشت (هر دو اصل ◊A↔¬ ¬A و A↔¬◊¬A را می پذیرفت). وی همچنین همتایی برای S4 ارائه داد و معناشناسی جبری و مدل متناهی را برای آن ارائه کرد.

در ۱۹۶۵ پراویتز[۱۱۶] همتای شهودگرایانه ای برای S4 و S5 را به همراه دستگاه استنتاج طبیعی شان ارائه داد. بعد ها در ۱۹۹۳ بیرمن[۱۱۷] و پیوا[۱۱۸] [2] سیستم او را برای S4 از نو و بدون ◊ فرمول بندی کردند تا نرمال سازی ساده تری به دست آید. آنان همینین به نظریه ی اثبات[۱۱۹] آن معناشناسی قطعی ای ارائه کردند که رویکرد جدیدی به استنتاج طبیعی در این منطق ها شد. این رویکرد دوباره و در سال ۱۹۹۲ بدون عملگر توسط بنوایدز[۱۲۰] و میباوم[۱۲۱] پیشنهاد شد. تعمیمی از این روش را ماسینی در سال ۱۹۹۳ شرح داد.

اونو[۱۲۲] به شیوه سنتی، مقایسه شهودگرایانه بین S4 و S5 ادامه داد. او بسیاری از متغیر های نا هم ارز مخصوصا بخشهای[۱۲۳] مستقل از ◊ را در دو منطق تجزیه و تحلیل کرد. او با ارایه معانی جبری و روش کریپکی خاصیت متناهی مدل را اثبات کرد. بعضی از این منطق ها بعدا توسط فونت[۱۲۴]

---

[114] Minc

[115] Bull

[116] Prawitz

[117] Bierman

[118] Paiva

[119] Proof Theory

[120] Benevides

[121] Maibaum

122 Ono

123 Fragment

124 Font

با جزییات بیشتر مورد تجزیه و تحلیل قرار گرفتند و موجهات آنها طبقه بندی شد. برای مثال ترکیب و ¬. (مقایسه شهودی S4، دارای ۳۱ ترکیب ناهم ارز اینگونه است. مثالی که در دوسن[۱۲۵] و میهایلوا[۱۲۶] نیز به آن اشاره شده است. در مقاله بعدی فونت و وردو[۱۲۷] صفات جبری مجرد (دوباره مستقل از ◊) را در مقایسه شهودگرایانه S4 و S5 بررسی کردند.

فیشر سروی[۱۲۸] در اولین مقالات خود (۱۹۷۷) روشی برای بدست آوردن مقایسه شهودی درست برای موجهات منطق کلاسیک پیشنهاد کرده است. او با استفاده از ترجمه گودل[۱۲۹] بر منطق شهودگرایانه به S4 کلاسیک، پیشنهاد کرد که یک منطق موجهات شهودی با ترجمه ای مشابه به منطق بای مودال کلاسیک قابل بیان است. منطقی با دو شرط . که یکی از منطق موجهات اصلی به ارث رسیده و دیگری شرط S4 برای شبیه سازی عملگر های شهودی. او معنا شناسی جهان های ممکن و جبری را در منطقی شهودی که با این شکل از منطق موجه کلاسیک بدست آمده ایجاد کرد. در مقاله سوم (۱۹۸۴) او برای اثبات کامل بودن نظریه اش، اصول بعضی از این منطق ها را با استفاده از معانی جهان واقع ارایه کرد. چهارچوب و قواعد مورد نظر او مستقلا توسط پلاتکین[۱۳۰] و استرلینگ[۱۳۱] (۱۹۸۶) بدست آمد.

اوالد[۱۳۲] در پایان نامه خود دز سال ۱۹۷۸ رویکردی فلسفی به منطق شهودی زمانی اتخاذ کرد. او بر پایه مدل کریپکی از منطق شهودی مرتبه اول و تفسیری از ایده هوشمندانه براور[۱۳۳] که در آن

125 Došen

126 Mihajlova

127 Verdu

128 Fischer Servi

129 Godel

130 Plotkin

131 Stirling

132 Ewald

133 Brouwer

ترتیب شناخت را از ترتیب زمانی جدا شده، مدل هایی برای منطق شهودی زمانی ارایه کرد. او اصول منطق موجهات شهودی را متناظر با وضعیت ترتیب زمانی یافت، و نتایج کامل بودن آن را اثبات کرد. او همچنین شکل دیگری از معناشناسی برای بدست آوردن خصوصیات مدل متناهی عرضه کرد. گرچه اثبات اوالد برای مدل متناهی غلط از آب درآمد، مدل های تصمیم پذیری که او ارایه کرد با مدل های جهان واقع فیشر و پلاتکین-استرلینگ یکسان است. اوالد، فیشر و پلاتکین-استرلینگ هر کدام جداگانه معانی یکسان و اصول کامل متناظر آن در منطق موجهات را پیشنهاد کردند.

گرچه چهارچوب معنایی آنها تنها گزینه نبوده، نظریه ای که برای این مدل توسط گابی، اونو، سوتیروف، بوزیک و دوسن، یوکوتا و ویسسکرا همگی با یکدیگر و همچنین با نظریه اوالد متفاوت اند، بسیاری مشابهات در میان آنها می توان یافت. نکته قابل توجه اینست که معناشناسی که توسط اونو و سوزوکی (۱۹۸۸) درباره آن بحث شده اگر چه در کل متفاوت از مدل فیشر است ولی با یک تبدیل مناسب برای S5 با این مدل یکسان خواهد بود.

در اینجا مختصرا به قوانینی اشاره می کنیم که در باید به هر یک از روش های ارائه شده در این پایان نامه افزوده شود تا این قوانین بتوانند برای بعضی از انواع منطق های موجهات شهودگرایانه به کار روند:

۳،۵. استنتاج خودکار در منطق های موجهات شهودگرایانه

روش تابلو: برای اینکه از روش تابلو در منطق‌های موجهات شهودگرایانه استفاده کنیم باید قوانین زیر را به آن بیافزاییم [1] :

قواعد ویژه‌ی عملگرهای شهودی:

$$[\mathrm{S}\neg]\ \frac{\Gamma, F\neg A}{\Gamma^{\#}, TA} \qquad (۵. ۱)$$

$[S\rightarrow]$ (۵. ۲)

روش رشته ای: بر اساس [1] اگر به قوانین فصل ۴ شامل قوانین کلاسیک و موجهات این قوانین را بیافزاییم منطق های موجهات شهودگرایانه‌ی نامبرده را نیز توسط روش رشته ای پیاده سازی کرده‌ایم.

برای سیستم‌های ID، IKDB، IKD4، IKD5 و IKD45:

$$[v]\ \frac{\Gamma \rightarrow \Delta}{\Gamma^{v} \rightarrow \Delta^{\pi}}$$ (۵. ۳)

برای سیستم‌های IT، IB، IS4 و IS5:

$$[L\quad]\ \frac{\Gamma, A \rightarrow \Delta}{\Gamma,\quad A \rightarrow \Delta}$$ (۵. ۴)

$$[R\Diamond]\ \frac{\Gamma \rightarrow \Delta, A}{\Gamma \rightarrow \Delta, \Diamond A}$$ (۵،۵)

پیوست ها

پیوست ۱: نکاتی فنی درباره‌ی پرانتزها و نمادگذاری

در پیاده سازی نرم افزار ملحق شده به این پروژه، از پرانتزها و نمادهای جاگذاری برای تفکیک زیررشته ها و نیز تشخیص دامنه ی عمل توابع به منظور تجزیه ی رشته های داده شده و ساخت درخت نحو آنها استفاده ی زیاد شده است. در این باره بد نیست که با نکاتی فنی در باره ی پرانتزها و قراردادهای ما در جداسازی رشته ها با استفاده از آنها آشنا باشیم:

• پرانتزها نمادهای کمکی نامیده می‌شوند. چنانکه دیدیم در درجه‌ی پیچیدگی فرمول تأثیری ندارند. آن‌ها همچنین در پیچیدگی درخت نحو هم اثری ندارند.

• تعداد پرانتزهای باز ″)″ با تعداد پرانتزهای بسته ″(″ در یک فرمول درست‌ساخت برابر است.

• اگر فرمول P بخش آغازی یک فرمول‌درست ساخت باشد، تعداد پرانتزهای باز **)** آن از تعداد پرانتزهای بسته **(** اکیدا بیش‌تر است.

• اگر فرمول P بخش پایانی یک فرمول درست‌ساخت باشد، تعداد پرانتزهای باز **)** آن از تعداد پرانتزهای بسته **(** اکیدا کمتر است.

نتیجه: بخش آغازین یا پایانی یک فرمول‌درست ساخت، خود نمی‌تواند یک فرمول درست‌ساخت باشد.

• تا بدین جا با تعریفی که از فرمول‌های درست‌ساخت ارائه کردیم، هر فرمول غیر اتمی باید در یک جفت پرانتز احاطه شده باشد. برای سادگی می‌توانیم این یک جفت پرانتز بیرونی را حذف کنیم. مثلا به جای $((A\wedge B)\rightarrow(\neg C))$ بنویسیم $(A\wedge B)\rightarrow(\neg C)$.

• گاهی اوقات برای سادگی و خوانایی بیش‌تر همه‌ی پرانتزها را از یک فرمول درست‌ساخت حذف می‌کنیم. این کار یگانه خوانی را از بین برده و موجب ابهام در تعبیر فرمول می‌شود. برای رفع این ابهام دو راه حل وجود دارد:

ترتیب اولویت عملگرها: یک فرمول بدون پرانتز می‌تواند *مبهم*[134] باشد، چرا که دامنه‌ی عمل رابط‌های منطقی آن ابهام دارد و می‌تواند به روش‌های گوناگون خوانده شود و در نتیجه فرمول‌های مختلفی از آن برداشت شود. به طور دقیق حذف پرانتزها موجب می‌شود نتوان تشخیص داد که کدام یک از عملگرها زودتر باید اعمال شوند. برای رفع این اشکال بین عملگرها اولویت قائل می‌شویم. بنا بر قرارداد اولویت عملگرها به این ترتیب است:

۱. ع

ملگرهای داخل پرانتز، عملگرهای درونی ترین پرانتزها همیشه اولویت دارند.

۲. $\neg$

۳. $\wedge$

$\vee$,

۴. $\rightarrow$

۵.

$\leftrightarrow$

134 Ambiguous

برای نمونه فرمول $A\wedge B\rightarrow\neg C$ را می توان صورت یک فرمول درست‌ساخت بازنویسی کرد. این فرمول ابهام دارد چون مشخص نیست که مثلا باید به کدام شکل از اشکال $(A\wedge B)\rightarrow(\neg C)$، $A\wedge[B\rightarrow(\neg C)]$ و یا $\neg[(A\wedge B)\rightarrow C]$ خوانده شود. با اعمال ترتیب اولویت عملگرها، این فرمول به صورت اول خوانده می شود.

نمادگذاری لهستانی: راه‌هایی وجود دارند که بدون ایجاد ابهام در تفسیر فرمول درست‌ساخت می‌توان پرانتزها را حذف کرد. برای این کار باید فرمول‌ها از نو و با ترتیبی دیگر نوشته شوند: کافی است تنها در عملگرهای دودویی به جای $A*B$ بنویسیم $*AB$ و از هیچ نمادکمکی مانند پرانتزها استفاده نکنیم. این نوع نگارش به افتخار ریاضیدان لهستانی *لوکاسیه‌ویچ*[۱۳۵] چنین نامیده شده‌است. در مقابل نگارش معمولی فرمول‌های درست‌ساخت که معادل *پیمایش میانوند*[۱۳۶] درخت دودویی نحو است، نمادگذاری لهستانی معادل *پیمایش پیشوندی*[۱۳۷] این درخت می‌باشد. در این نگارش هیچ ابهامی در تخصیص دامنه‌ی عمل رابطهای منطقی پیش نمی‌آید و نیازی هم به تعیین اولویت بین عملگرها نیست. برای نمونه فرمِ لهستانی فرمول $\neg A\rightarrow B$ عبارت است از $\rightarrow\neg AB$. برای فرمول $\neg(A\rightarrow B)$ عبارت است از $\neg\rightarrow AB$. برای فرمول $A\rightarrow\neg B$ عبارت است از $\rightarrow A\neg B$. و برای فرمول $\neg((A\wedge\neg B)\vee(C\rightarrow D))\wedge B$ نیز عبارت است از $\wedge\neg\vee A\neg B\rightarrow CDB$.

در واقع می توان نمایش لهستانی و معمولی فرمول‌های درست‌ساخت را به وساطت درخت نحو به یکدیگر تبدیل نمود. در غیر این صورت باید دامنه‌ی عمل را مشخص کرد.

135 Lukasiewicz

136 Infix

137 Prefix

پیوست ۲: قضیه ی یگانه خوانی[138]

قضیه: هر فرمول درست‌ساخت مانند X از $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$، دقیقا در یکی از پنج شرط زیر صدق می‌کند:

- X نمادهای صفر موضعی صدق یا تناقض ($\perp$ یا $\top$) است.
- X یک گزاره‌ی اتمی (مانند A، B، C و ...) است.
- X به صورت $(\neg P)$ است که P یک فرمول درست‌ساخت یکتا از $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ است.
- X دقیقا به یکی از صورت های $(P\wedge Q)$، $(P\vee Q)$، $(P\rightarrow Q)$، $(Q\leftarrow P)$، $(P\leftrightarrow Q)$ است، به طوری که P و Q فرمول های درست‌ساخت یکتا از $L(\mathbf{R}, \mathbf{F}, \mathbf{C})$ هستند.

از قضیه‌ی یگانه خوانی در می‌یابیم که نمی‌توان یک فرمول درست‌ساخت از منطق مرتبه‌ی صفر را به دو معنی تفسیر کرد:

- علاوه بر یکسان بودن دامنه‌ی عمل رابط های منطقی یک موضعی و دو موضعی، دامنه‌ی عمل سورها نیز یکتا است.
- درخت نحو در منطق مرتبه‌ی صفر یکتا است.

[138] Unique Parsing Theorem

پیوست ۳: درخت نحو[139]

درخت نحو: با توجه به تعاریفی که از ترم، اتم و فرمول ارائه شد، فرآیند تولید فرمول های درست‌ساخت را می‌توان به طور ساده جمع بندی نمود:

- *تشکیل ترم ها:* ترم‌ها توسط نمادهای تابعی ترکیب شده و نرم‌های نو می‌سازند.
- *تشکیل اتم ها:* ترم‌های نهایی توسط رابطه‌ها ترکیب شده و اتم‌ها را می‌سازند.
- *تشکیل گزاره های درست‌ساخت:* هر اتم می‌تواند یک گزاره‌ی درست‌ساخت باشد. با ترکیب گزاره های درست‌ساخت با یکدیگر توسط رابط های گزاره‌ای و یا افزودن سور بر سر آنها می‌توان گزاره های درست‌ساخت پیچیده تری تولید کرد.

مانند منطق گزاره‌ها می‌توان مراحل ترکیب فرمول های کوچکتر را به منظور ساختن فرمول های بزرگتر، به صورت *درخت نحو*[140] نمایش داد. شکل زیر قوانین تشکیل درخت نحو را بر اساس تعریف فرمول درست‌ساخت در منطق مرتبه صفر نشان می‌دهد. برگ‌ها گزاره های اتمی یا توابع صفرموضعی هستند و در درخت نحو به ترم‌های تشکیل دهنده شان تجزیه نمی‌شوند.

$\perp$

اتم‌ها و رابط های ۰ تایی برگ اند. ($\perp$)

رابط یک موضعی، یک فرزند دارد. ($\neg$A)

O

A B

رابط دودویی، دو فرزند دارد. (AOB)

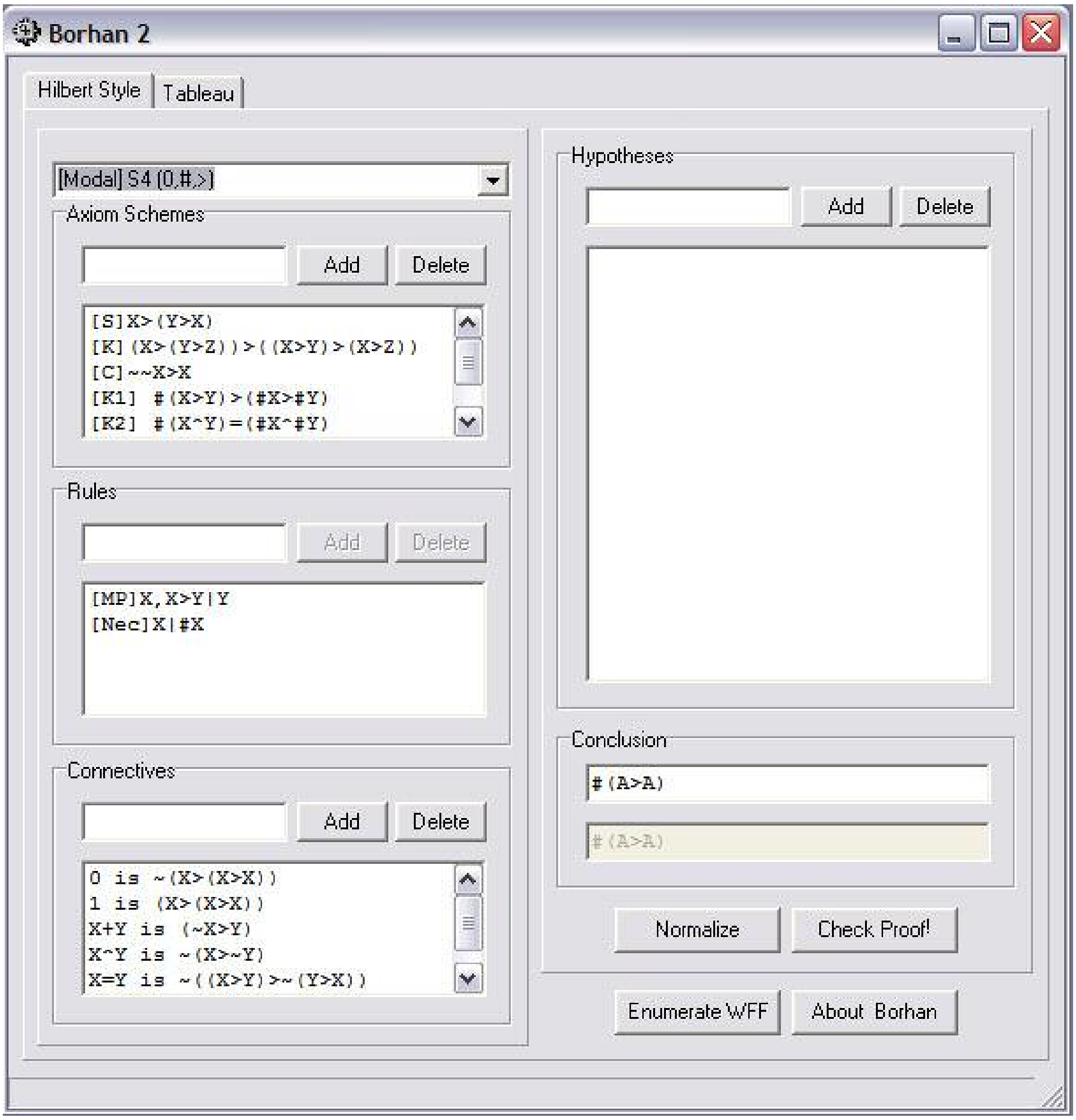
شکل ۱ (پیوست). قوانین ساختن درخت نحو برای منطق مرتبه‌ی صفر

[139] Unique Parsing Theorem

[140] Syntax tree

پیوست ۴: نمونه هایی از پیاده سازی

روش‌هایی که در این پروژه ذکر شده اند توسط نرم افزاری به زبان دلفی پیاده‌سازی شده اند که لوح فشرده‌ی آن به پایان نامه پیوست شده است:



شکل ۳ (پیوست): واسط کاربری روش اصل موضوعی هیلبرتی

چندان که در شکل صفحه ی قبل مشاهده می کنید، واسط کاربری روش اصل موضوعی از دو ستون تشکیل شده است. ستون چپ منطق استنتاج را تعریف می کند و ستون سمت راست حکمی را که باید قابل استنتاج بودن آن تحقیق شود با فرضیات احتمالی اش دریافت می کند.

در ستون سمت چپ، برای معرفی هر منطق باید سه گروه اصول موضوعه، قواعد و تعاریف عملگرها وارد شود:

۱. اصول موضوعه با قالب زیر تعریف می شوند:

[*Axiom Name***]** *Axiom Scheme as well-formed formulas (containing* **X***,* **Y***,* **Z***)*

برای نمونه اصل موضوع [K1] $\square(X\rightarrow Y)\rightarrow(\square X\rightarrow\square Y)$ به صورت زیر وارد می شود:

```
[K1] #(X>Y)>(#X>#Y)
```

۲. قواعد با این قالب وارد می شوند:

[*Rule Name***]** *Hypothesis 1***,** *Hypothesis 2***,** *... (all containing* **X***,* **Y***,* **Z***, ...)* **|** *Conclusion containing* **X***,* **Y***,* **Z***, ...*

مثلا قوانین Modus Ponens و Nec در منطق های موجهات به صورت زیر برای برنامه تعریف می شوند:

```
[MP] X,X>Y | Y
[Nec] X | #X
```

عاریف عملگرها با این قالب وارد می شوند:

Zeroary, Unary or Binary Connective (may be containing **X, Y***)* **is** *a well-formed formula (containing* **X, Y***)*

برای نمونه اگر در اصول موضوعه‌ی دستگاه ما عملگرهای $\rightarrow$ و $\neg$ پوشش داده شده‌اند، اما اصل موضوعی برای تعریف $\wedge$ وجود ندارد می توان $X\wedge Y$ را به صورت $\neg(X\rightarrow\neg Y)$ تعریف کرد و بنا بر آن تعریف عملگر $\wedge$ را بر مبنای عملگرهای دیگر به این صورت وارد کرد:

```
X^Y is ~(X>~Y)
```

این امکان وجود دارد که بعد از ورود هر سه گروه منطق تعریف شده را به عنوانی ذخیره کرد. در شکل صفحه ی قبل منطق موجهات S4 را می بینید که عملگرهای $\neg$ و $\rightarrow$ در آن با اصول موضوعه تعریف شده است و از این رو نام آن را `[Modal]S4(~,#,>)` گذاشته ایم. در حال حاضر ۳۵ دستگاه مختلف شامل کلاسیک، موجهات، شهودگرایانه، فازی و خطی برای این برنامه از پیش تعریف شده اند (شکل ۴ – پیوست).

در ستون سمت راست در قسمت بالا پیش فرض های مورد نظر به صورت فرمول های درست ساخت وارد می شوند. در قسمت پایین هم یک فرمول درست ساخت به عنوان حکم وارد می شود که هدف نهایی ما تحقیق این است که آیا از پیش فرض های داده شده و در چارچوب منطق تعریف شده بر اساس اصول و قواعد و تعاریف مشخص) قابل استنتاج است یا نه. بنا بر قرار داد فرض و حکم ها را بر اساس گزاره های اتمی با حروف A، B، C، و ... وارد می کنیم که با حروف X، Y و ... اکسیوم ها که گزاره ی اتمی نیستد و هر یک می توانند یک فرمول مرکب باشند، تداخل پیدا نکنند.

با استفاده از گزینه‌ی Normalize نرم افزار اقدام به بازنویسی فرمول داده شده می‌کند به نحوی که اگر در آن از عملگری استفاده شده که در اصل موضوع های دستگاه پوشش داده نشده است، بر اساس تعریف عملگر جدید، آن را بر اساس عملگرهای تعریف شده در اصول بازنویسی می کند. مثلا در نمونه ی ارائه شده برای این دستگاه هر جا $A\wedge B$ را دید عبارت $\neg(A\rightarrow\neg B)$ را به جای آن می نشاند.

در پایان با انتخاب گزینه ی Check Proof! نرم افزار شروع به کار می کند. به این نحو که تمام احکام استنتاج شده در این دستگاه را یک به یک استنتاج می کند تا به حکم ما برسد. چندان که در فصل ۲ هم توضیح داده شده است. این روش در صورت قابل استنتاج نبودن حکم پیشنهادی نمی

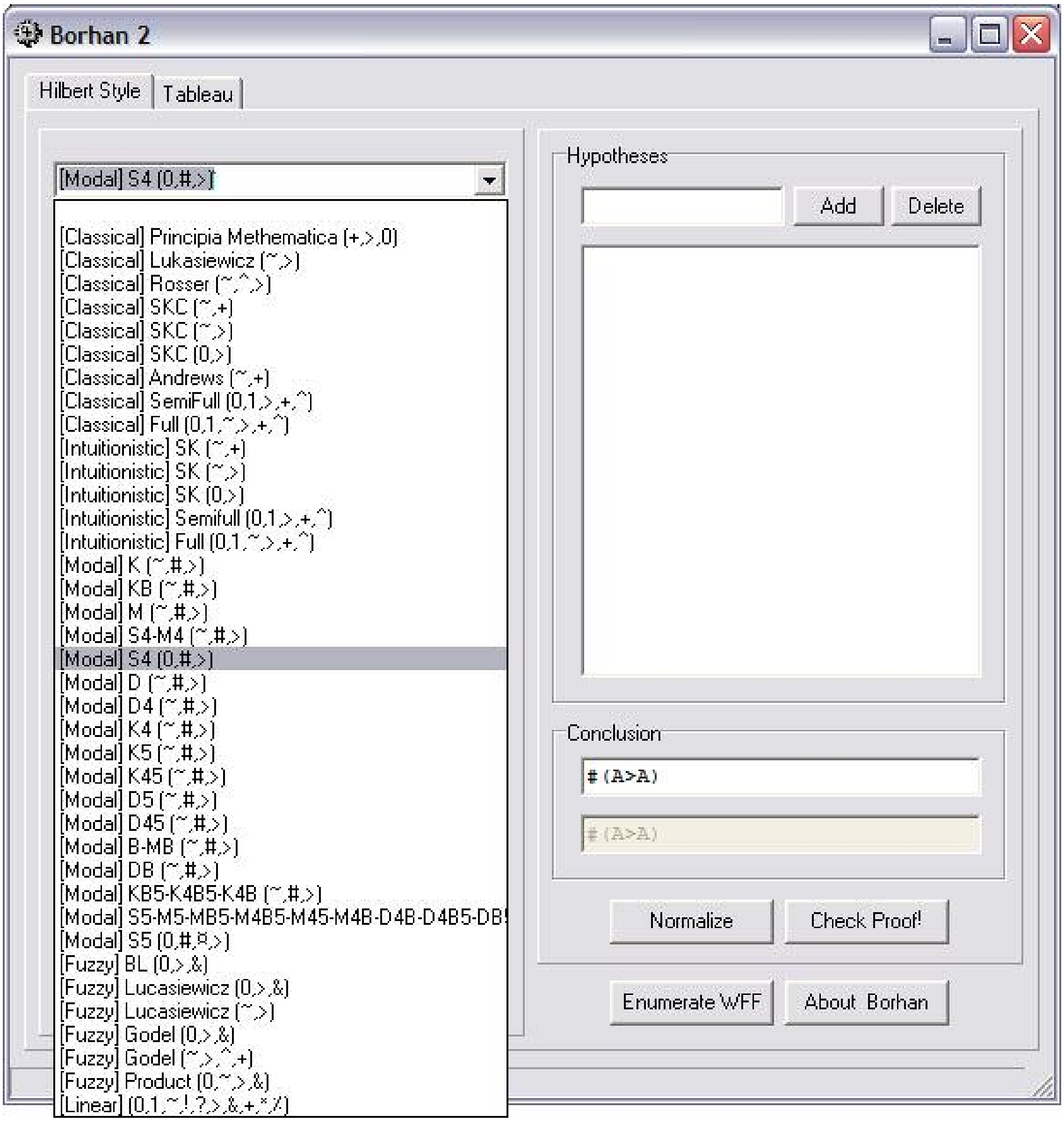
تواند ما را آگاه کند. مادامی که نرم افزار در حال جستجو است هر دو احتمال وجود دارد: حکم داده شده قابل استنتاج نیست. و یا اینکه این حکم قابل استنتاج است ولی دستگاه ما هنوز به آن نرسیده است. پس مساله ی صادق بودن یا نبودن حکم در اینجا بازگشتی شمارشی[۱۴۱] است و نه بازگشتی[۱۴۲] [4].

در صورتی که برنامه در فهرست فرمول های استنتاج شده اش به حکم پرسش شده برخورد کند، با یک الگوریتم عقبگرد[۱۴۳] گزاره هایی که در اثبات این فرمول نقش نداشته اند را حذف می کنند و نقش فرمول های موثر را نیز با ذکر شماره ی آنها مشخص می سازد. برای نمونه به برهان حکم $(A \rightarrow A)$ توجه کنید (شکل ۵ – پیوست).

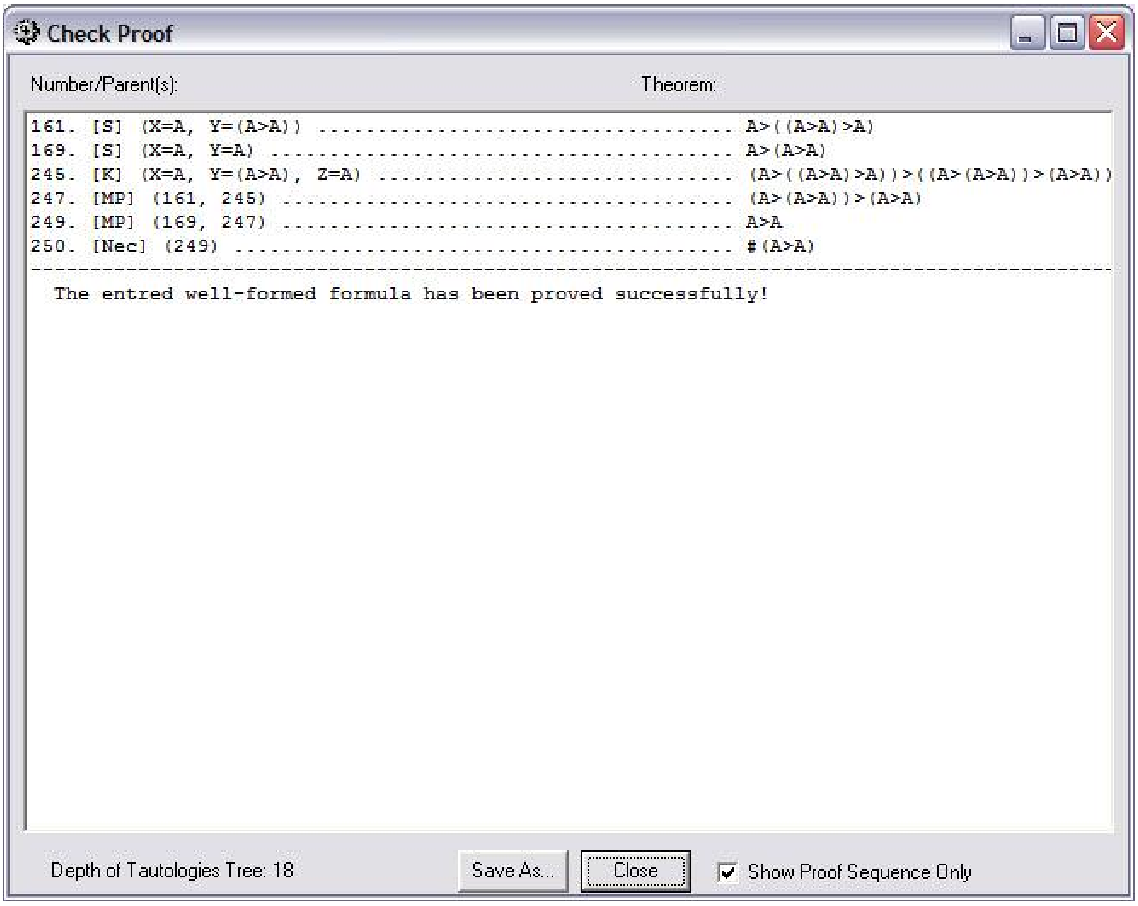
141 Recursively Enumerable

142 Recursive

143 Back Track



شکل ۴ (پیوست): دستگاه های اصل موضوعی مرتبه صفر تعریف شده برای نرم افزار



شکل ۵ (پیوست): برهان برای فرمول $(A\rightarrow A)$

قسمت دیگر پیاده سازی شده در این نرم افزار روش تابلو است. در اینجا هم قواعد تابلو به قالب دقیقی به برنامه داده می شوند:

[*Rule Name***] G,** *Signed formula (over the line)* /
G, *Signed formula1***,** *Signed formula2*, ... **| G,** *Signed formula1***,** *Signed formula2*, ...

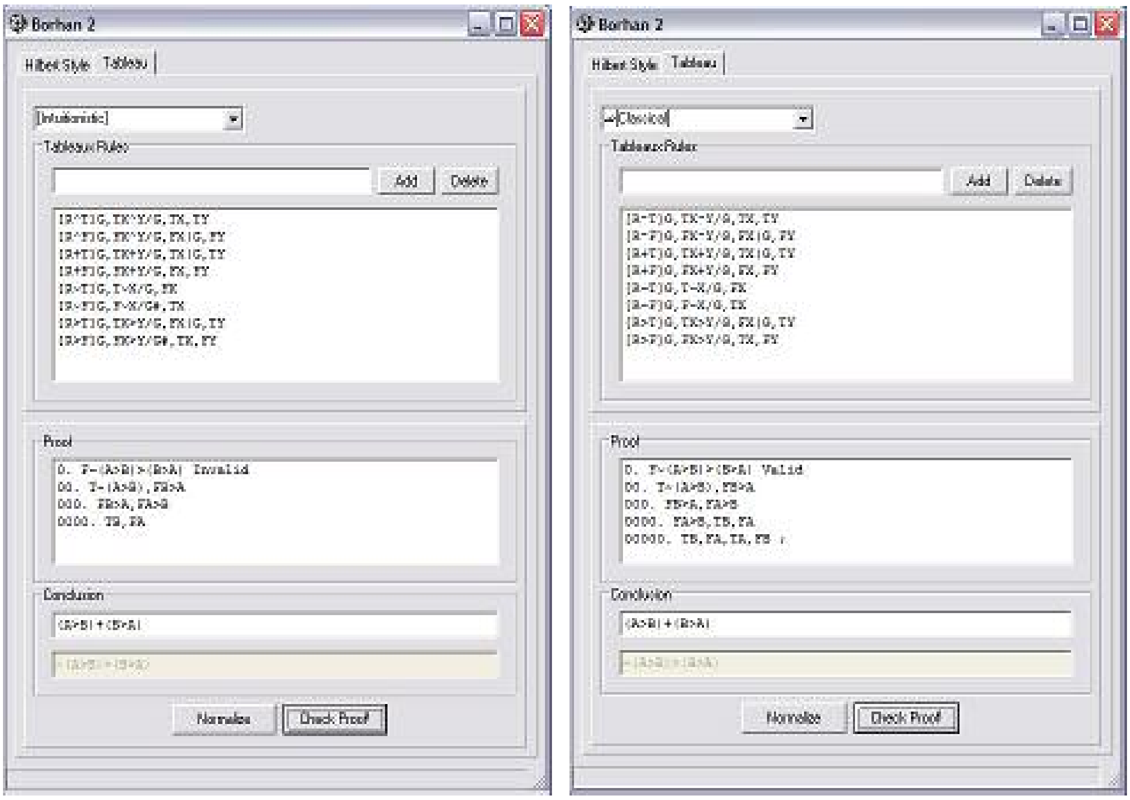
برای نمونه می توانیم قواعد تابلوی زیر را

$$[\text{R}\wedge\text{T}]\ \frac{\Gamma, TA \wedge B}{\Gamma, TA, TB} \qquad\qquad [\text{R}\wedge\text{F}]\ \frac{\Gamma, FA \wedge B}{\Gamma, FA \mid \Gamma, FB}$$

به این صورت برای نرم افزار تعریف کنیم:

```
[R^T] G,TX^Y/G,TX,TY
[R^F] G,FX^Y/G,FX|G,FY
```

شکل ۶ (پیوست) نشان می دهد که چگونه یک فرمول یکسان در دو دستگاه مختلف شهودگرایانه و کلاسیک تابلو های مختلف دارد و در نتیجه در یکی معتبر و در دیگری نامعتبر است:



شکل ۶ (پیوست): روش تابلو نشان می دهد که فرمول (A→B)∨(B→A)

در منطق کلاسیک معتبر است، اما در منطق شهودگرایانه نامعتبر

منابع:

[1] Amati G. et al., *A Uniform Tableau Method for Intuitionistic Modal Logics*, Italian PT Administration and Fondazione Ugo. Bordoni, 1993

[2] Bierman G.M. et. al, *on an Intuitionistic Modal Logic*, Kluwer Academic Publishers, Netherlands,1999

[3] Bridges D. S., *Computability, A mathematical Sketchbook,* BridgesSpringer-Verlag, 1994

[4] Cypser M., *Introduction to the Theory of Computation,* PWS Publishing Company, 1997

[5] Dyer C. R., *First-Order Logic (CS 540 Lecture Notes)*, University of Wisconsin – Madison.

[6] Fitch F. B., *Intuitionistic modal logic with quantiers*, Portugaliae Mathem aticae, 7:113-118, 1948

[7] Fitting M., *First-Order Logic & Automated Theorem Proving,* Springer.

[8] Fitting M., *Proof methods for modal and intuitionistic logics*, Reidel, Dordrecht, 1983

[9] Fitting M., *Tableaus for Many-Valued Modal Logic*, Research partly supported by NSF, 1995

[10] Goubault-Larrecq J., *Proof Theory and Automated Deduction*, Kluwer Academic Publishers, 1997

[11] http://en.wikipedia.org/wiki/Semantics_of_modal_logic

[12] Linz P., *An Introduction Forward Languages and Automata,* Jones and Bartlet Publishers, 2001

[13] Papadimitriou C. H., *Computational Complexity,* Addison-Wesley Publishing Company, 1994

[14] Prior A. N., *Time and Modality*, Oxford, 1957

[15] Simpson A. K., *The Proof Theory and Semantics of Intuitionistic Modal Logic*, University of Edinburgh

[16] Shen W., *Automated Proofs of Equivalence of Modal Logic Systems*, Thesis for Master of Science, University of Miami, 2006