دانشکده علوم ریاضی

پیشنهاد موضوع تحقیق پایان‌نامه کارشناسی ارشد

نام و نام خانوادگی:

شماره دانشجوئی:

رشته و گرایش: ریاضی محض (منطق ریاضی)

عنوان به فارسی:

عنوان به انگلیسی: Permission to Speak: A Logic for Access Control and Conformance

سابقه، اهداف، نوع، کاربردهای مورد نظر (با ذکر آخرین مراجع):

مراحل، روش و برنامه زمان‌بندی تحقیق (با تاکید بر جنبه‌های جدید کار):

امکانات مورد نیاز، نحوه تامین و برآورد هزینه:

سایر موارد:

مالکیت نتایج:

کلیه حقوق مادی مترتب بر نتایج تحقیق پایان‌نامه متعلق به دانشگاه است و انتشار نتایج نیز تابع مقررات دانشگاهی است و با موافقت استاد راهنما صورت می گیرد.

امضاء دانشجو:

تاریخ:

تایید استاد (استادان) راهنمای پایان‌نامه:

در صورت تصویب موضوع تحقیق پیشنهاد بدینوسیله آمادگی خود را برای راهنمائی و مشاوره دانشجو در کلیه مراحل انجام و ارائه تحقیق و مشارکت در ارزیابی پایان‌نامه و بر اساس ضوابط دانشگاه و دانشکده اعلام و افراد زیر را جهت تعیین استاد مشاور پیشنهاد می‌نمایم (می‌نمائیم)

۱–　　۲–　　۳–

نام و امضای (استاد استادان) راهنما　　**تاریخ:**

نظر کمیته تحصیلات تکمیلی دانشکده:

پایان‌نامه پیشنهادی در جلسه تحصیلات تکمیلی دانشکده علوم ریاضی مورخ

□ تصویب شد و　　به عنوان استاد مشاور تعیین گردید.

□ تصویب نشد.

نام و نام خانوادگی سرپرست تحصیلات تکمیلی دانشکده: دکتر اعظم اعتماد

امضاء

تاریخ:

جلسه دفاع از پایان‌نامه کارشناسی ارشد

اجازه سخن گفتن: منطق کنترل دسترسی و هماهنگی قانونی

سخنران: محمد سبحانیان

زمان: چهارشنبه ۱۵ تیرماه ۱۳۹۰ ساعت ۱۰:۳۰صبح

مکان: سالن خوارزمی، دانشکده ریاضی

هیئت داوران

۱ – دکتر مجتبی آقایی

۲ – دکتر محمدرضا رئوفی

۳ – دکتر رسول رمضانیان(دانشگاه صنعتی شریف)

۴ – دکتر اعظم اعتماد

چکیده:

متن چکیده

دانشگاه صنعتی اصفهان

دانشکده علوم ریاضی

اجازه سخن گفتن: منطق کنترل دسترسی و هماهنگی قانونی

پایان‌نامه کارشناسی ارشدریاضی محض (منطق ریاضی)

محمد سبحانیان

استاد راهنما

دکتر مجتبی آقایی

۱۳۹۰

دانشگاه صنعتی اصفهان

دانشکده علوم ریاضی

پایان‌نامه کارشناسی ارشدریاضی محض (منطق ریاضی)آقای محمد سبحانیان

تحت عنوان

اجازه سخن گفتن: منطق کنترل دسترسی و هماهنگی قانونی

در تاریخ ۱۵ تیرماه ۱۳۹۰ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهائی قرار گرفت.

۱– استاد راهنمای پایان‌نامه	دکتر مجتبی آقایی
۲– استاد مشاور پایان‌نامه	دکتر محمدرضا رئوفی
۳– استاد داور ۱	دکتر رسول رمضانیان (دانشگاه صنعتی شریف)
۴– استاد داور ۲	دکتر اعظم اعتماد

سرپرست تحصیلات تکمیلی دانشکده دکتر اعظم اعتماد

تقدیم به پدرم و همسر عزیزم

فهرست مطالب

چکیده:

در این پایان نامه به صورت‌بندی و بیان یک سیستم استنتاجی برای یک منطق پرداخته‌ایم که هم برای کنترل دسترسی و هم برای بررسی تطابق با قوانین مفید است. این سیستم منطقی قابلیت بیان سیستم‌های کنترل دسترسی، به عنوان یک دستگاه حقوقی و قانونی را دارد. نخست عملگر $says$، از منطق‌های کنترل دسترسی را مورد توجه قرارداده‌ایم. عملگر $says$ نخست به عنوان یک وجه در یک منطق موجهاتی بیان می‌شود. در این دیدگاه عملگر $says$ مفهوم «گفتن» را به صورت شهودی مدل‌سازی می‌کند. سپس با بیان آن براساس حساب لاندای نوع‌دار و بررسی سیستم استنتاجی و اصول موضوعه‌ی آن در حساب لاندای‌نوع‌دار، نشان داده شده که سیستم استنتاجی شهودگرایانه، بر استدلال بر محرمانگی، با توجه به سطوح محرمانگی، مطابقت دارد. برای عملگر $says$ سه تعبیر «درخواست»، «تأیید» و «اعلان» را معرفی شده‌است. از جمله مفاهیم مهمی که با استفاده از عملگر $says$ قابل بیان است، «وکالت» و «نقل‌قول» هستند. سپس عملگر $says$ را با بردن به سیستم‌های حقوقی تکمیل کرده و به آن امکان داده شده که بنا به کابرد آن در جمله، تعبیر شود. سیستم‌های حقوقی برای بیان و صورت‌بندی قوانین و بیان سیستم استنتاجی مربوط به آن ارائه شده‌اند. سیستم‌های حقوقی به علت پیچیدگی‌های خاص خود ما را به سیستم‌های استنتاجی غیریکنوا راهنمایی می‌کنند. در این پایان‌نامه سیستم‌استنتاجی مورد نظر را به تفصلی بیان‌کرده‌ایم که با استفاده از ارزش «نامشخص» برای برخی جملات، توانمند شده‌است. در سیستم‌های استنتاجی غیریکنوا ممکن است چند جواب، از یک مجموعه فرضیات بدست آید. علاوه براین در «اجازه سخن گفتن» با استفاده از وظایف تودرتو توانسته‌ایم قابلیت‌های زبانی بالایی را در اختیار بگیریم. در این منطق «وظیفه» و «مجوز» به عنوان دو نوع وجه که دوگان یکدیگر هستند، معرفی شده‌اند و این به ساختار زبانی و استنتاجی معمولی بشری بسیار نزدیکتر است. با بیان چند مثال قابلیت‌های زبانی و استنتاجی ارائه شده را تشریح کرده و در نهایت با بیان و بررسی پارادکس پروتاگوراس، نشان داده‌ایم که سیستم منطقی معرفی شده قابلیت بررسی سیستم‌های حقوقی پیچیده را نیز دارد. در این پایان نامه به ارائه یک سیستم منطقی به همراه اصول موضوعه مناسب آن پرداخته شده که با مدل‌های کریپکی معناشناسی شده وبرای آن اثبات قضایای سلامت و تمامیت نیز ارائه شده. و در نهایت با استفاده از زیرفرمول‌ها تصمیم‌پذیری را نیز برای آن نشان داده‌ایم.

کلمات کلیدی: اجازه سخن گفتن، منطق تکلیف، استنتاج غیریکنوا، قوانین، پرونده‌های حقوقی، کنترل دسترسی، استدلال شکاک، استدلال ساده‌لوح

فصل ۱

مقدمه

کنترل دسترسی عمری طولانی در تفکر بشری دارد، اما از ورود آن به منطق چندان نمی‌گذرد. به خصوص استفاده از عملگر $says$ (گفتن) که به عنوان یک وجه برای کنترل دسترسی در سال ۱۹۹۳ توسط مارتین ابدی [۵] مطرح شد. وی با ارائه‌ی عملگر $says$، کنترل دسترسی را به منطق موجهاتی برد و یک حساب برای آن معرفی کرد. در این مقاله عملگر $says$ بنا به درک شهودی پایه‌گذاری شده است. وی با توجه به مسئله‌ی امنیت و محرمانگی، که در [۳] نیز به آن پرداخته، محرمانگی و تکنیک کلیدهای عمومی را وارد و سعی کرده نشان دهد عملگر $says$ نیازهای محرمانگی در کنترل دسترسی را نیز برآورده می‌کند. علاوه براین وی با بیان انواعی از «وکالت»، «نقل قول» و «سخن گفتن از طرف دیگری»، قابلیت‌های بالای این منطق را نشان داده و در نهایت با به‌کاربردن یک عملگر for توانسته یک دستگاه استنتاجی با ویژگی نرمال‌فرم برای آن بسازد که در آن استنتاج بسیار ساده تر است.

همچنین، وی در سال ۱۹۹۸ در [۳۶] به برنامه نویسی برای بحث امنیت و یک‌پارچگی پرداخته و قواعد استنتاجی برای آن معرفی نموده‌است. این حساب استنتاجی بر اساس حساب لاندا و البته حساب لاندای نوع‌دار پایه‌گذاری شده است. وی به هر عامل یک سطح از محرمانگی اختصاص می‌دهد و با فرض یک ترتیب جزئی روی عامل‌ها، این سطح‌بندی را بیان می‌کند. بدین طریق محرمانگی و یک‌پارچگی را برای منطق به‌دست آمده، به ارمغان می آورد.

در سال ۲۰۰۷ مارتین ابدی در [۱] بحث محرمانگی و کارکرد آن، وکالت و نقل قول را در حساب لاندا، بر اساس کارهای قبلی نظیر [۴] و [۵] بیان می‌کند. وی این دو منطق و حساب را با هم تلفیق کرده

و یک حساب استنتاجی در حساب لاندا برای کنترل‌دسترسی پایه‌ریزی می‌کند که در آن عملگر $says$ که در [۵] بر اساس درک شهودی از کنترل دسترسی و مفهوم «گفتن» ساخته شده بود، با حساب هسته‌ی وابستگی که بر اساس حساب لاندای نوع‌دار و سطوح محرمانگی [۴]، بیان می‌شود و برخی قواعد استنتاج که در [۵] به عنوان اصل پذیرفته می‌شوند را به عنوان قضیه نتیجه می‌گیرد. در این مقاله (یعنی [۱]) ملاحظه می‌شود که درک شهودی ارائه شده برای مفهوم «گفتن» با آن‌چه از حساب وابستگی و حساب محرمانگی به‌دست آمده، تطابق دارد. علاوه بر این، با اضافه کردن سیستم F و چند ریختی DCC به حساب لاندای نوع‌دار، قابلیت‌های استنتاجی زبان را به منطق‌های مرتبه دوم گسترش می‌دهد. بدین ترتیب می‌شود کنترل دسترسی را به عامل‌هایی که دارای خاصیت خاصی هستند، اعطا کرد.

عملگر $says$ که توسط مارتین ابدی ابداع شده است را باید چگونه تعبیر نمود؟ در حالت ساده گوییم «A می‌گوید s». اما این «گفتن» به چه منظور است؟ فرض کنید $test(x)$، تست هپاتیت باشد. در این صورت $A\ says\ test(B)$ را به «A تست هپاتیت روی B را می‌گوید» تعبیر می‌کنیم. اما این یعنی چه؟ یعنی «A می‌خواهد که روی B تست هپاتیت انجام شود»؟ یا «A می‌گوید روی B تست هپاتیت انجام شده است»؟ عملگر $says$ را می‌توان به سه طریق توصیف کرد:

۱) به عنوان درخواست

۲) به عنوان تأیید

۳) به عنوان اعلان

در یک سیستم کنترل دسترسی معمولاً از مورد اول استفاده می‌شود. یعنی $A\ says\ s$ به معنای «عامل A نیاز s را درخواست می‌کند» تعبیر می‌شود. در فصل ۳ نیز بیشتر همین جنبه مدنظر قرار گرفته است. اما در فصل ۳ مشاهده می‌کنیم که s را می‌توانیم به عنوان یک فرمول در نظر بگیریم. در این صورت $A\ says\ s$ به صورت دقیق‌تر به «عامل A درستی فرمول s را درخواست کرده» تعبیر می‌شود. اما حال که s را یک فرمول در نظر گرفته‌ایم، می‌توانیم $A\ says\ s$ را به عنوان «تأیید» نیز تعبیر کنیم. در این صورت $A\ says\ s$ به معنای «عامل A درستی فرمول s را تأیید می‌کند» خواهد بود.

با دو مثال زیر این تفاوت را آشکار می‌کنیم:

مثال: فرض کنید در یک دانشکده، یک رئیس دانشکده (A) داریم و یک مسئول بایگانی پرونده‌ها (B)، که به پرونده‌های دانشکده (r) دسترسی دارد. زمانی یک شخص به پرونده‌ای دسترسی دارد که مسئول بایگانی (B) بخواهد. به عبارتی، زمانی $access(A,r)$ برآورده می‌شود که B بگوید $access(A,r)$.

بنابراین گوییم B بر دسترسی A به r کنترل دارد و می‌نویسیم $B\ contrlos\ access(A,r)$. بنابراین:

$$B\ says\ access(A,r) \Rightarrow access(A,r)$$

اما چون A رئیس دانشکده است، پس اگر به B دستور بدهد که پرونده‌ای را به او بدهد، B باید اطاعت کند. بنابراین:

$$A\ says\ access(A,r) \Rightarrow B\ says\ access(A,r)$$

پس با داشتن $A\ says\ access(A,r)$ خواهیم داشت $access(A,r)$. یعنی اگر A دسترسی به r را تقاضا کند، این دسترسی اتفاق خواهد افتاد.

در کنترل دسترسی، مسأله همین است. شخصی مانند A تقاضایی را درخواست می‌کند و باید تصمیم بگیریم که آیا این تقاضا باید برآورده شود یا خیر. در صورتی نیاز اظهار شده، برآورده می‌شود که به صورت استنتاجی به دست آید.

از دیگر قابلیت‌های این سیستم، بررسی توانایی افراد در دسترسی داشتن بر نیازها است. به طور مثال، دیدیم که رئیس دانشکده نیز بر دسترسی به پرونده‌ها کنترل دارد و می‌تواند دسترسی داشته باشد و می‌تواند این دسترسی را به هرکسی که بخواهد، اعطا کند.

مثال: فرض کنید یک دانشکده داریم و یک استاد. نمره درس یک دانشجو زمانی اعتبار دارد که دانشگاه آن را تأیید نماید. دانشگاه زمانی نمره‌ی یک دانشجو را تأیید می‌کند که دانشکده نمره وی را تأیید نماید و دانشکده نمره‌ی دانشجو را به استاد مربوطه واگذار کرده است.

در این‌جا، اگر استاد درس ریاضی نمره دانشجو در آن درس را تأیید نماید، دانشکده و دانشگاه نیز آن نمره را تأیید می‌کنند. البته فرمول‌بندی آن ساده بوده و مشابه مثال قبل است. مفاهیم مربوط به صورت‌بندی این مسائل را در فصل ۳ به تفصیل بیشتری بیان کرده‌ایم.

منطق ارائه شده برای هر دو تعبیر «درخواست» و «تأیید» کارا بوده و هر دو تعبیر را مدل‌سازی می‌کند. اما تعبیر سوم (اعلان) با این منطق مدل‌سازی نمی‌شود. این تعبیر، در فصل ٦ برای عملگر $says$ بیان می‌شود. البته پیش از آن، در فصل ۵ مفهومی از اعلان بیان شده است. در فصل ٦ عملگر $says$ بنا به فرمولی که بعد از آن می‌آید و بنا به فرمولی که $says$ در آن قرار گرفته و همچنین با توجه به کاربرد آن در کل سیستم، می‌تواند هر یک از سه مفهوم ارائه شده را به خود اختصاص دهد.

در فصل‌های ۳ و ۴ به عملگر $says$ یک عامل اصلی به عنوان فاعل آن نسبت داده می‌شود، اما در فصل ٦ فرمول $A\ says\ \varphi$ به $says_{l(A)}\varphi$ تغییر می‌کند. به علاوه در این فصل از نماد $\mathcal{O}_A\varphi$ برای نشان دادن «موظف بودن عامل A، به برآورده شدن فرمول φ»، استفاده می‌کنیم. فرمول $says_{l(A)}\mathcal{O}_B\varphi$ یعنی:

«تقاضای برآورده شدن فرمول φ توسط عامل A، از عامل B»، و همچنین اگر $\mathcal{P}_A\varphi$ را به معنای «B مجاز است که ...» در نظر بگیریم، $says_{l(A)}\mathcal{P}_B\varphi$ به معنای «A به B اجازه می‌دهد که فرمول φ برآورده شود» خواهد بود. همان‌طور که مشاهده می‌کنید در تعبیر این نوع جملات، به جای «برآورده کند» از «برآورده شود» استفاده کرده‌ایم. این بدان معنا است که تعبیر معمولی و روزمره‌ی انسانی از این گونه جملات با درک این زبان از این جملات کمی متفاوت است.

توجه داریم که در این فصل، یک عامل، تقاضای خود را، از عاملی خاص درخواست می‌کند.

در مثال‌های فصل ٦ مشاهده می‌شود که عملگر $says$ می‌تواند معنای تأیید نیز بدهد. اما گاهی نیز فقط معنای اعلان دارد. نه تأیید می‌کند و نه درخواست.

مفهوم اعلان در فصل ۵ پایه‌ریزی شده است. عملگرهای by_{Id} از فصل ۵ و $says_{Id}$ از فصل ٦ بسیار شبیه به هم رفتار می‌کنند. هرگاه $says_{Id}$ فقط معنای اعلان بدهد، همانند by_{Id} رفتار می‌کند. در عملگر $says$، مجموعه‌ی Id، مشخص کننده‌ی یک عامل به عنوان فاعل عمل گفتاری $says$ می‌باشد، اما در by_{Id} فقط به مجموعه‌ای از قوانین اشاره دارد. این تفاوت کلیدی به ما اجازه نمی‌دهد که در فصل ۵، به جای by_{Id} از $says_{Id}$ استفاده کنیم.

در سیستم‌های حقوقی و قانونی، اولین چیزی که جلب توجه می‌کند صورت‌بندی قوانین است. قوانین را به صورت $id : \varphi \mapsto \psi$ صورت بندی می‌کنیم که این نمادگذاری هم در فصل ۵ و هم در فصل ٦، برای قوانین به کار رفته است. اما دیدگاه ما در دو فصل ۵ و ٦ به قوانین یکسان نیست و به دو گونه متفاوت تعبیر می‌شوند.

در فصل ٦ هر قانون به یک عامل منحصر به فرد اشاره دارد اما درفصل ۵ فقط یک مجموعه از قوانین داریم. در فصل ٦ یک عامل، عامل دیگری را «موظف» کرده یا «مجوزی» برای او صادر نموده است. اما در فصل ۵ این سیستم حقوقی است که یک شخص را موظف به کاری می‌کند. البته توجه داریم که در فصل ۵ قوانین برای همه یکسان هستند، اما در فصل٦ می‌توانیم قوانینی وضع کنیم که مختص یک عامل خاص باشد. البته در فصل ۵ نیز با استفاده از محمول‌های کمکی می‌توان این کار را انجام داد.

بنابراین در دیدگاه فصل ۵، قانون $id.\mathbf{o} : \varphi \mapsto \psi$ می‌گوید، بعد از جای‌گذاری عامل‌ها در متغیرها (که با تابع تخصیص صورت می‌گیرد)، «هرگاه φ برآورده شود، باید ψ نیز برآورده شود». توجه داریم که این «باید» همزمان است و به آینده اشاره ندارد، اما با منطق زمانی ـ که در فصل ۵ به آن اشاره شده ـ می‌توانیم این مهم را به انجام برسانیم. اما به عبارتی دقیق‌تر، «هرگاه گزاره‌ی $\nu(\varphi)$ برآورده شود، قانون id، گزاره‌ی $\nu(\psi)$ را اعلان می‌کند». اما در فصل ۵ قوانین از دو نوع «وظیفه» و «مجوز» هستند. اگر قانون از نوع «وظیفه» باشد، اعلان آن یک وظیفه را مشخص می‌کند. اما اگر از نوع «مجوز» باشد، فقط یک اعلان به ما می‌دهد که می‌تواند در بیان مقدم قوانین مورد استفاده قرار گیرد.

توجه داریم که «وظیفه» بر عهده‌ی سیستم می‌باشد است و این سیستم است که موظف به برآورده شدن ψ است و اگر نقض شود، یک خطا برای سیستم گزارش می‌شود.

در فصل ٦، قانون «$id : \varphi \mapsto \psi$» دارای یک شرط اضافی است. شناسه‌ی id متعلق به قوانین یک عامل منحصربه‌فرد است. در این صورت قانون «$id : \varphi \mapsto \psi$» را به صورت «اگر φ برآورده شود، عامل A ـ که $id \in l(A)$ ـ بنا به قانون id گزاره‌ی ψ را اعلان می‌کند» تعبیر می‌کنیم. در این‌جا، اعلان کردن را با استفاده از عملگر $says$ نشان می‌دهیم. بنابراین فاعل این عمل گفتاری مشخص است.

این نکته در بحث هماهنگی با قوانین، خود را به بهترین شکل نشان می‌دهد. در فصل ۵، هماهنگی با قوانین برای کل سیستم تعریف می‌شود و باید همه‌ی عامل‌ها به همه‌ی قوانین پای‌بند باشند، در حالی که در فصل ٦ هماهنگی یک عامل با یک عامل معنا دارد. در فصل ٦ می‌گوییم «عامل A هماهنگ با قوانین عامل B رفتار کرده است، اگر هر وظیفه‌ای که عامل B برای A اعلان کرده باشد، به وسیله‌ی A برآورده شده باشد».

در فصل پنجم به طور مستقیم با استفاده از عملگرهای $\mathcal{O}_y$ (وظیفه) و $\mathcal{P}_y$ (مجوز) به منطق تکلیف گام نهاده‌ایم. در منطق تکلیف، یک عامل می‌تواند موظف به یک «فعل» یا «عمل» باشد ولی در مورد اعلان کردن باید یک گزاره را اعلان کند. بنابراین در فرمول‌های $\mathcal{P}_y\varphi$ و $\mathcal{O}_y\varphi$، فرمول φ باید یک «عمل» را نشان دهد و در فرمول $says_{l(A)}\varphi$، فرمول φ می‌باید یک گزاره باشد. حال سؤال مهمی مطرح می‌شود که باید به آن پاسخ داد: محمول‌ها از نوع «عمل» هستند یا از نوع «گزاره»؟ دو راه به ذهن می‌رسد:

راه اول: محمول‌ها را به دو دسته تقسیم می‌کنیم، برخی «فعل» و برخی «گزاره» هستند. در این صورت، در فرمول $says_{l(A)}p(x)$، محمول $p(x)$ یک «گزاره» بوده و فرمول $says_{l(A)}p(x)$ یک «عمل» را نشان می‌دهد و همچنین در فرمول $\mathcal{O}_Aq(y)$، عبارت $q(y)$ یک عمل را نشان می‌دهد و فرمول $\mathcal{O}_Aq(y)$ یک گزاره را به نمایش می‌گذارد.

هرچند این راه بسیار قوی و قابل قبول بوده و قابلیت‌های قابل‌توجهی را به سیستم منطقی اعطا می‌کند، اما به دلیل پیچیدگی‌هایی که به دنبال دارد از آن صرف نظر می‌کنیم.

راه دوم: در یک دیدگاه ساده می‌توانیم همه را گزاره در نظر بگیریم. در این صورت $says_{l(A)}\varphi$ را به صورت «عامل A بنا به قوانین $l(A)$، گزاره‌ی φ را اعلان کرده است» تعبیر می‌کنیم و فرمول $\mathcal{O}_Asays_{l(A)}\varphi$ را به صورت «عامل A موظف است که بنا به قوانین $l(A)$ فرمول φ را اعلان کرده باشد».

به سادگی می‌توانیم از یک عملگر act استفاده کنیم که اگر p یک عمل باشد، $act(p)$ یک گزاره باشد به معنای «عمل p انجام شده است». بنابراین act از یک عمل یک گزاره می‌سازد و می‌توانیم

ادعا کنیم برای سادگی در نگارش فرمول‌ها، از act صرف نظر کرده‌ایم. البته مؤلفان ۲۱ به این مسئله بی‌توجه بوده‌اند، اما می‌توانیم امیدوار باشیم که عملگر act محدودیتی به سیستم اِعمال نکند و البته بررسی آن را به آینده وامی‌گذاریم.

در فصل ۳ که به بررسی عملگر $says$ پرداخته شده، با دیدگاهی موجهاتی به آن نگاه کرده‌ایم و استنتاج در این سیستم، یکنوا است. البته مارتین ابدی سعی کرده با بررسی عملگر $says$ قابلیت‌های آن در مدل‌سازی مفاهیمی مانند «وکالت»، «سخن گفتن از طرف دیگری» و «نقل قول» را نشان دهد. وی با استفاده از تکنیک کلیدهای عمومی نشان داده که عملگر $sayss$، از استنتاج‌هایی که در گذشته، در بحث حفظ محرمانگی مطرح بوده، نیز حمایت می‌کند.

مارتین ابدی با به کاربردن حساب لاندای نوع‌دار و بیان سیستم استنتاجی عملگر $says$ بر اساس آن، عملگر $says$ را از حالت شهودی خارج کرده و نشان داده که با بحث‌های گذشته در مورد کنترل دسترسی و محرمانگی اطلاعات هم‌خوانی دارد. وی با به کارگیری سیستم F، قدرت استنتاجی سیستم ارائه شده را از منطق‌های مرتبه اول به منطق‌های مرتبه دوم توسعه داده و با استفاده از چندریختی‌های DCC موفق به انجام این امر شده است. البته ما بحث چندریختی‌های DCC را به اختصار بیان کرده‌ایم و از پرداختن جدی به آن خودداری کرده‌ایم.

استنتاج ارائه شده در فصل ۵ از نوع غیریکنوا می‌باشد و ممکن است با اضافه شدن به فرضیات، برخی از نتایج قبلی را از دست بدهیم. پایه‌ی استنتاج غیریکنوا در منطق قراردادی است و در آن‌جا است که بیشتر بر روی آن کار شده است. البته کارهای دیگری که در استنتاج غیریکنوا صورت گرفته را می‌توان در استنتاج بر سیستم‌های منطق شناختی جستجو کرد. لازم به ذکر است که دستگاه استنتاجی ارائه شده در فصل‌های ۵ و ٦ از منطق قراردادی قوی‌تر بوده و تمام قابلیت‌های آن را شامل می‌شود. به خصوص منطق ارائه شده در فصل ٦.

در این سیستم‌ها فرضیاتی به همراه قواعدی برای استنتاج داریم، و اصول موضوعه‌ای که از استنتاج گزاره‌ای ساده در آن زبان پشتیبانی می‌کنند. همان‌طور که در فصل‌های ۵ و ٦ مشاهده می‌شود، برای این سیستم‌ها، جهان‌هایی متعدد، ممکن است که در آن‌ها قوانین و فرضیات برآورده شده باشند.

این نوع از استنتاج، حالتی قوی‌تر از آن‌چه تا کنون، در هوش مصنوعی به کار رفته، ایجاد می‌کند و می‌تواند راه‌کاری مناسب برای برنامه‌نویسی سیستم‌های استنتاجی پیچیده باشد و می‌توان از آن برای تحلیل سیستم‌های پیچیده‌ی انسانی بهره برد.

فصل ۲

پیش‌نیازها

برای مطالعه این پایان نامه به مفاهیمی از قبیل شبکه‌ها، نیم شبکه‌ها و نیم‌شبکه‌ـ نیم‌گروه‌ها نیاز داریم که در بخش‌های اول و دوم این فصل به آن‌ها می‌پردازیم. این ساختارها برای بررسی سیستم‌های کنترل دسترسی سلسله مراتبی مورد نیاز هستند. در ادامه با سیستم F از حساب لاندای نوع‌دار آشنا می‌شویم که برای سیستم‌های کنترل دسترسی توزیع شده در فصل ۳ مورد استفاده قرار می‌گیرند و به ما امکان داشتن منطق‌های مرتبه دوم را می‌دهند.

در این فصل همه مطالب به صورت خلاصه بیان شده و فقط به جهت برطرف کردن نیاز خواننده برای مطالعه این پایان نامه می‌باشد. البته اگر می‌خواستیم بیشتر از این به بیان مطالب بپردازیم می‌بایست کتابی قطور را به عنوان پیشنیازها ارائه می‌نمودیم. البته سعی شده در متن پایان نامه نیز از مطالبی که پیچیدگی زیادی دارند پرهیز کنیم. البته این به سبب وسعت مطالب می‌باشد و اگر بخواهیم به طور تخصصی هر یک از این مطالب را باز کنیم خود کاری بزرگتر از این پایان‌نامه را طلب می‌کند.

۲ — ۱ شبکه‌ها

دو ساختار مجزا برای شبکه‌ها معرفی می‌کنیم که در انتها خواهیم دید این دو دیدگاه هم‌ارز هستند، یکی دیدگاه ترتیبی که به شبکه به عنوان یک ترتیب جزئی نگاه می‌کند و دیگری دیدگاه جبری است که به یک شبکه به عنوان یک ساختار جبری می‌نگرد. برای آشنایی با مطالب این بخش و دیدن اثبات قضایا (که

البته غالبا سرراست و ساده هستند)، می‌توانید به [۱۵] رجوع کنید.

۲–۱–۱ شبکه‌ها به عنوان یک ترتیب جزئی

تعریف ۱.۲ فرض کنیم $(L, \leq)$ یک ترتیب جزئی باشد. $(L, \vee, \wedge)$ یک شبکه است اگر دو شرط زیر را برآورده سازد:

۱) برای هر $a, b \in L$ کوچکترین کران بالای مجموعه‌ی $\{a, b\}$ وجود دارد که الحاق a و b نامیده شده و با $a \vee b$ نشان داده می‌شود.
یعنی برای هر $c \in L$ که $a \leq c$ و $b \leq c$ داریم $a \vee b \leq c$.

۲) برای هر $a, b \in L$ بزرگترین کران پایین مجموعه‌ی $\{a, b\}$ وجود دارد که انجمن a و b نامیده شده و با $a \wedge b$ نشان داده می‌شود.
یعنی برای هر $c \in L$ که $c \leq a$ و $c \leq b$ داریم $c \leq a \wedge b$.

تساوی به شکل معمول تعریف می‌شود یعنی $a = b$ اگر و تنها اگر $a \leq b$ و $b \leq a$.

گزاره ۲.۲ روی هر شبکه ترتیبی، اعمال الحاق ($\vee$) و انجمن ($\wedge$) هر دو دارای خواص شرکت‌پذیری، جابجایی و خودتوانی هستند.

یک شبکه را کران‌دار گوییم، اگر دارای بزرگترین و کوچکترین عضو باشد. معمولاً بزرگترین عضو را با $\top$ و کوچکترین عضو را با $\bot$ نشان می‌دهیم و هر شبکه دلخواه را می‌توان با اضافه کردن دو عضو $\top$ و $\bot$ به آن، به یک شبکه کراندار تبدیل کرد.

تعریف ۳.۲ فرض کنیم $A \subseteq L$ و $(L, \leq)$ یک شبکه باشد. در این صورت تعریف می‌کنیم:

- $\bigwedge A = \bigwedge_{a \in A} a$
- $\bigvee A = \bigvee_{a \in A} a$

و به عنوان قرارداد تعریف می‌کنیم $\bigvee \emptyset = \bot$ و $\bigwedge \emptyset = \top$.

قضیه **۴.۲** فرض کنید $(L, \leq)$ یک شبکه باشد. با اضافه کردن $\top$ و $\bot$ به L می‌توانیم از آن یک شبکه کراندار بسازیم.

قضیه **۵.۲** فرض کنید $(L, \leq)$ یک شبکه بوده و A و B دو زیرمجموعه از L. در این صورت داریم:

۱) $\bigvee(A \cup B) = (\bigvee A) \vee (\bigvee B)$

۲) $\bigwedge(A \cup B) = (\bigwedge A) \wedge (\bigwedge B)$

۳) $\bigvee(A \cup \emptyset) = (\bigvee A) \vee (\bigvee \emptyset) = (\bigvee A) \vee \bot = \bigvee A$

۴) $\bigwedge(A \cup \emptyset) = (\bigwedge A) \wedge (\bigwedge \emptyset) = (\bigwedge A) \wedge \top = \bigwedge A$

دو عبارت آخر سازگاری تعریف با $A \cup \emptyset = A$ را نشان می‌دهد.

تذکر **۶.۲** قضیه فوق برای $\cap$ برقرار نیست.

قضیه **۷.۲** اگر $(L, \leq)$ یک شبکه ترتیبی باشد، عملگرهای انجمن $(\wedge)$ و الحاق $(\vee)$ را می‌توان به شکل زیر تعریف کرد:

۱) برای هر $a, b \in L$ داریم: $a \leq b$ اگر و تنها اگر $a = a \wedge b$.

۲) برای هر $a, b \in L$ داریم: $a \leq b$ اگر و تنها اگر $b = a \vee b$.

۲–۱–۲ شبکه‌ها به عنوان یک ساختار جبری

تعریف **۸.۲** ساختار جبری $(L, \wedge, \vee)$ را یک شبکه گوییم هرگاه:

۱) $\vee$ شرکتپذیر، جابجایی و خودتوان باشد.

۲) $\wedge$ شرکتپذیر، جابجایی و خودتوان باشد.

۳) برای هر $a, b \in L$ داشته باشیم $a \vee (a \wedge b) = a$ و $a \wedge (a \vee b) = a$