دانشگاه اراک

دانشکده فنی و مهندسی

کارشناسی ارشد مهندسی کامپیوتر گرایش نرم‌افزار

ارائهٔ روشی رسمی مبتنی بر سناریوها برای وارسی سیستم‌های توصیف شده توسط سیستم تبدیل گراف

دانشجو

کیانوش رشیدی

استاد راهنما

دکتر وحید رافع

استاد مشاور

دکتر محسن رحمانی

شهریور ۱۳۹۲

به نام خدا

ارائهٔ روشی رسمی مبتنی بر سناریوها برای وارسی سیستم‌های توصیف شده توسط سیستم تبدیل گراف

توسط:

کیانوش رشیدی

پایان نامه

ارائه شده به مدیریت تحصیلات تکمیلی به عنوان بخشی از فعالیت‌های تحصیلی لازم برای اخذ درجه کارشناسی ارشد در رشته مهندسی کامپیوتر- گرایش نرم‌افزار

از

دانشگاه اراک

اراک- ایران

ارزیابی و تصویب شده توسط کمیته پایان نامه با درجه:.....

دکتر وحید رافع(استاد راهنما).....استادیار

دکتر رضا رافع(داور).....استادیار

دکتر محسن رحمانی(استاد مشاور).....استادیار

شهریور ۱۳۹۲

به نام خدا

تقدیر و تشکر

با تشکر فراوان از آقایان دکتر وحید رافع و دکتر محسن رحمانی که با راهنمایی‌های سازنده و مفید خود، مرا در روند انجام این پایان‌نامه یاری کردند. همچنین، با سپاس فراوان از پدر و مادر مهربان، دلسوز و عزیزم که همیشه در تمام مراحل زندگی یار و پشتیبان من بوده و هستند.

چکیده

با افزایش پیچیدگی روز افزون سیستم‌های نرم‌افزاری، صحت‌سنجی سیستم‌ها نیز امری دشوارتر و پیچیده‌تر می‌شود و این امر در مورد سیستم‌هایی که در کاربردهای حساس مورد استفاده قرار می‌گیرند از اهمیت بیشتری برخوردار خواهد بود. از اینرو، نیاز به روش‌هایی که در عین سادگی در کاربرد، با قابلیت اطمینان بالا صحت و درستی سیستم را تضمین کنند احساس می‌شود. یکی از روش‌هایی که می‌تواند صحت و درستی سیستم را با قابلیت اطمینان بالا تضمین کند، روش وارسی مدل می‌باشد. از طرف دیگر، سیستم تبدیل گراف فُرمالیسم گرافیکی است که به عنوان روشی موثر و کارا در امر مدل‌سازی سیستم‌ها شناخته می‌شود. در نتیجه می‌توان با استفاده از عملیات وارسی مدل در زمینهٔ سیستم‌های تبدیل گراف، یک چارچوب مناسب و کارا در زمینهٔ بررسی صحت و درستی سیستم‌های نرم‌افزاری فرآهم کرد. با این حال، در کاربردهای واقعی، تکنیک وارسی مدل با مشکل انفجار فضای حالت روبرو می‌باشد و همین امر باعث می‌شود که نتوان از سودمندی این روش در امر بررسی فرمال سیستم‌ها به طور کامل بهره برد. از طرف دیگر، روش‌های فُرمالی مانند منطق زمانی (LTL، CTL، TCTL، CTL*...) که برای بیان خصوصیات، بمنظور بررسی آن‌ها در سیستم‌ها مورد استفاده قرار می‌گیرند، در برخی موارد در کاربردهای واقعی ناکارامند می‌باشند. زیرا در همهٔ موارد نمی‌توان خصوصیات سیستم را در قالب فرمول‌های منطقی بیان کرد، و به این منظور نیاز به روش‌های طبیعی‌تری می‌باشد.

در کار پیش رو با توجه به مشکلات بیان شده در امر بررسی سیستم‌ها، سعی بر ارائه روشی شده که با استفاده از توانایی سناریوها در بیان خصوصیات رفتاری سیستم، امکاناتی به منظور بررسی سیستم‌هایی که با استفاده از سیستم تبدیل گراف مدل شده‌اند فرآهم شود. با تکیه بر سناریوهای رفتاری می‌توان به جای جستجو کردن کل فضای حالت سیستم، فقط آن بخش از فضای حالت را گسترش داد که به رفتار مشخص شده در سناریو مربوط می‌باشد. در نتیجه فضای حالت به طور چشمگیری کاهش خواهد یافت و امکان بررسی خصوصیاتی که با روش‌های رایج قابل بررسی نبود، فرآهم خواهد شد. مزیت دیگر استفاده از سناریوها در امر بررسی سیستم‌ها سادگی آن‌ها در بیان خصوصیاتی است که با استفاده از منطق زمانی به دشواری بیان می‌شدند.

کلمات کلیدی: نمودار Live Sequence Chart، تجزیه و تحلیل مبتنی بر سناریو، کاهش فضای حالت، سیستم تبدیل گراف

فهرست مطالب

فصل اول – مقدمه ۱

۱ – ۱ – شرح مختصری از بحث ۲

۱ – ۲ – اهداف پایان‌نامه ۵

۱ – ۳ – ساختار پایان‌نامه ۶

فصل دوم – مروری بر ادبیات موضوع ۷

۲ – ۱ – سیستم تبدیل گراف ۷

۲ – ۱ – ۱ – گراف نوع و گراف میزبان ۹

۲ – ۱ – ۲ – قوانین تولید ۱۰

۲ – ۲ – سناریوها و نیازمندی‌های سیستم ۱۱

۲ – ۳ – نمودار LSC ۱۲

۲ – ۳ – ۱ – مولفه‌ها، پیغام‌ها و فعالیت‌های محلی ۱۳

۲ – ۳ – ۲ – مکان‌ها ۱۴

۲ – ۳ – ۳ – درجه حرارت ۱۴

۲ – ۳ – ۴ – شرط‌ها ۱۶

۲ – ۳ – ۵ – ناحیه‌های همزمانی و Co-regionها ۱۶

۲ – ۳ – ۶ – نمودار وجودی و نمودار جامع ۱۷

۲ – ۴ – بررسی سیستم‌های نرم‌افزاری و عملیات وارسی مدل ۱۸

۲ – ۴ – ۱ – خصوصیات زمان خطی ۲۳

۲ – ۴ – ۲ – منطق زمانی ۲۴

۲ – ۵ – ابزار Groove ۲۷

۲ – ۵ – ۱ – زبان کنترل در Groove ۲۸

فصل سوم – مروری بر کارهای مرتبط ۳۰

۳ – ۱ – عملیات وارسی مدل در زمینهٔ سیستم تبدیل گراف ۳۱

۳ – ۲ – روش‌های بهینه سازی و کاهش فضای حالت ۳۲
۳ – ۲ – ۱ – روش وارسی مدل نمادی ۳۳
۳ – ۲ – ۲ – روش کاهش ترتیب جزئی ۳۴
۳ – ۲ – ۳ – روش تقارن ۳۵
۳ – ۲ – ۴ – روش On-the-Fly ۳۶
۳ – ۳ – راهکارهای مبتنی بر سناریو ۳۷
فصل چهارم – ارائهٔ راهکار مبتنی بر سناریو در سیستم تبدیل گراف ۴۰
۴ – ۱ – ایدهٔ راهکار مبتنی بر سناریو درسیستم تبدیل گراف ۴۱
۴ – ۲ – معرفی ویرایشگر LSC ۴۶
۴ – ۳ – فرمال‌سازی سناریوها ۴۸
۴ – ۳ – ۱ – تعاریف فرمال مربوط به عناصر نمودار LSC ۴۸
۴ – ۳ – ۲ – سمانتیک رفتاری نمودار LSC ۵۲
۴ – ۴ – ترکیب سناریوها با سیستم تبدیل گراف ۵۴
۴ – ۵ – توصیف رفتار عناصر موجود در LSC در ابزار Groove ۶۳
۴ – ۵ – ۱ – پیغام‌ها و فعالیت‌های محلی ۶۳
۴ – ۵ – ۲ – نقیض رخدادها ۶۴
۴ – ۵ – ۳ – شرط‌ها ۶۵
۴ – ۵ – ۴ – ناحیه‌های هم‌زمان و co-regionها ۶۷
۴ – ۵ – ۵ – حلقه‌های تکرار ۷۰
فصل پنجم – بررسی نمونه‌های عملی ۷۲
۵ – ۱ – مسئلهٔ غذا خوردن فیلسوف‌ها ۷۲
۵ – ۲ – سیستم آژانس مسافرتی ۷۵
۵ – ۳ – بازی Pac-Man ۷۹
۵ – ۴ – سیستم دیوارهٔ آتش ۸۰

فصل ششم – جمع‌بندی و نتیجه‌گیری ۸۱
مراجع ۸۵
صفحه ی چکیده و صفحه عنوان به زبان انگلیسی

فهرست جداول

جدول (۵– ۱): نتایج بـدست آمده از بررسی خصوصیـات مورد نظر در مسئلۀ غذا خوردن فیلسوف‌ها ۷۵

جدول (۵ - ۲): نتایج بدست آمده از بررسی خصوصیات مورد نظر در مسئلۀ آژانس مسافرتی ... ۷۷

جدول (۵ - ۳): نتایج بدست آمده از بررسی خصوصیت مورد نظر در بازی Pac-Man ۷۹

جدول (۵ - ۴): نتایج بدست آمده از بررسی خصوصیت مورد نظر برای مسئلۀ دیوارۀ آتش..... ۸۰

فهرست شکل‌ها

شکل (۲ - ۱) : نمونه‌ای از نحوهٔ اجرای قانون p بر روی گراف G و تولید گراف H..... ۱۰

شکل (۲ - ۲) : نمونه‌ای از نمودار LSC که از نوع جامع تعریف شده ۱۳

شکل (۲ – ۳) : نمونه‌ای از آتاماتای کنترل ۲۹

شکل (۴ - ۱) : وضعیتی که مشتری اتاقی را رزرو کرده است و بدون این که هزینهٔ آن را پرداخت کرده باشد از سیستم خارج شده است ۴۲

شکل (۴ - ۲) : سناریویی بمنظور بررسی این که آیا مشتری می‌تواند اتاقی را رزرو کند و بدون این که هزینهٔ آن را بپردازد از سیستم قطع شده و خارج شود ؟ ۴۶

شکل (۴ - ۳) : tab مربوط به ویرایشگر LSC در ابزار Groove..... ۴۸

شکل (۴ - ۴) : سناریویی که در قالب نمودار LSC از نوع وجودی بیان شده به همراه برش‌های موجود در نمودار ۵۸

شکل (۴ - ۵) : آتاماتای معادل با نمودار LSC مشخص شده در شکل (۴- ۴) ۵۹

شکل (۴ - ۶) : آتاماتای معادل با فعالیت محلی ۶۳

شکل (۴ - ۷) : آتاماتای معادل با فعالیت محلی به همراه پیغام ۶۴

شکل (۴ - ۸) : آتاماتایی که رفتار معادل با عملگر نقیض یک رخداد را نشان می‌دهد ۶۵

شکل (۴ - ۹) : آتاماتای نشان‌دهندهٔ عنصر شرط در نمودار.....۶۶

شکل (۴ - ۱۰) : به کار گرفتن عملگر نقیض در عنصر شرطی و آتاماتای معادل با آن ۶۶

شکل (۴ - ۱۱) : آتاماتایی که رفتار معادل با سناریویی که حاوی یک co-region و یک ناحیه همزمان می‌باشد را نشان می‌دهد ۶۷

شکل (۴ - ۱۲) : نمودار LSC که نشان‌دهندهٔ استفاده از عملگر نقیض رخداد در ناحیهٔ همزمان و co-region می‌باشد ۶۸
شکل (۴ - ۱۳) : آتاماتایی که رفتار معادل با سناریوی شکل (۴- ۱۲) را نشان می‌دهد ۶۹
شکل (۴ - ۱۴) : آتاماتای معادل با سناریویی که دارای حلقهٔ تکرار while می‌باشد ۷۰
شکل (۴ - ۱۵) : آتاماتای معادل با سناریوایی که دارای ساختار حلقهٔ ALAP می‌باشد ۷۱
شکل (۵ - ۱) : سناریوی بیان کنندهٔ این که دو فیلسوف هم‌زمان نباید یک چنگال را در دست داشته باشند ۷۳
شکل (۵ - ۲) : سناریویی که نشان‌دهندهٔ خصوصیت بن‌بست در سیستم می‌باشد ۷۴
شکل (۵ - ۳) : الف) سناریوی بیان کنندهٔ موقعیتی که مشتری در صورت رزرو اتاق هتل نباید بدون پرداخت هزینه از سیستم خارج شود ب) بیان کنندهٔ این سناریو که آیا حالتی وجود دارد که مشتری پس از عملیات رزرو و پرداخت پول، پیام تصدیق از بانک دریافت نکند ۷۸
شکل (۵ - ۴) : سناریوایی که Pac-Man و Ghost نباید در خانهٔ مشخص شده همزمان حضور داشته باشند ۷۹
شکل (۵ -۵) : در صورت تولید یک بستهٔ غیر مطمئن، آن بسته نباید به هیچ ترتیبی از دیوارهٔ آتش عبور کند ۸۰

فهرست الگوریتم‌ها

جدول (۴– ۱): الگوریتم ادغام قوانین به منظور شبیه‌سازی نواحی هم‌زمان ۶۱

جدول (۴ - ۲): نحوهٔ اجرای سناریوها در سیستم تبدیل گراف ۶۱

فصل اول

مقدمه

امروزه، در بیشتر زمینه‌های زندگی استفاده از سیستم‌های کامپیوتری نقش اساسی را ایفا می‌کند و روزبه‌روز نیز بر میزان استفاده از آن‌ها افزوده می شود. علاوه بر افزایش و فراگیر شدن کاربرد سیستم‌های کامپیوتری، پیچیدگی آن‌ها نیز به طور مداوم رو به افزایش می‌باشد. از طرف دیگر، برخی از سیستم های کامپیوتری در کاربردهای حساسی مانند نیروگاه‌ها، پالایشگاه‌ها، سیستم‌های هوایی یا دستگاه‌های پزشکی و از این قبیل کاربردها که بروز خطا و اشکال در آن‌ها ممکن است منجر به ضرر و زیان‌های مالی و جانی جبران ناپذیری شود، مورد استفاده قرار می‌گیرند. در نتیجه باید به منظور بررسی درستی این سیستم‌ها از روش های دقیقی استفاده کرد که بالاترین قابلیت اطمینان را برای سیستم‌های کامپیوتری تضمین کنند. در کل هدف از بررسی سیستم های نرم افزاری[1] اطمینان از این موضوع می باشد که سیستم مورد نظر، بدون هیچ عیب و نقصی تمامی نیازمندی‌ها و مشخصات مورد انتظار کاربران را برآورده سازد. بحثی که به

1 - Software verification

عنوان محور اصلی این پایان‌نامه قرار گرفته نیز در زمینهٔ بررسی سیستم‌های نرم‌افزاری می‌باشد که در بخش‌های بعدی به جزئیات آن پرداخته شده است[1, 2].

۱ – ۱ – شرح مختصری از بحث

بررسی سیستم‌های نرم افزاری می تواند به روش های مختلفی انجام گیرد. برخی از روش های متداول در بررسی سیستم های نرم‌افزاری عبارت اند از: تست، شبیه سازی، نظارت و بازبینی مستندات و روش های فرمال. روش های فرمال در بررسی سیستم های نرم افزاری بر مبنای تئوری دقیق ریاضی بنا شده‌اند. به عبارت دیگر می توان روش های فرمال را استفاده از ریاضیات کاربردی بمنظور مدل‌سازی و تجزیه و تحلیل سیستم های نرم افزاری در نظر گرفت. معمول ترین روش فرمال بررسی سیستم های نرم افزاری وارسی مدل[1] می باشد که بر روی مدل سیستم اعمال می‌گردد. در این روش سیستم مورد نظر به همراه ویژگی(خصوصیت)[2] که باید در محصول نهایی موجود باشد بر اساس قواعد ریاضی مدل شده و سپس این دو مدل برای انجام عملیات بررسی به نرم‌افزار وارسی کنندهٔ مدل[3] تحویل داده می شوند و وارسی کنندهٔ مدل تشخیص می دهد که آیا ویژگی مورد نظر در سیستم وجود دارد یا خیر. به طور کلی عملیات وارسی مدل به این صورت انجام می شود که وارسی کنندهٔ مدل از روی مدل سیستم مورد نظر، کلیهٔ حالاتی را که سیستم می تواند در آن قرار گیرد را ساخته و ویژگی مورد نظر که معمولاً با استفاده از فرمالیسم منطق زمانی[4] بیان می‌شود را در تک‌تک حالات قابل حصول سیستم مورد بررسی قرار می‌دهد. به کلیهٔ حالات قابل حصول سیستم، فضای حالت[5] سیستم گفته می شود که از وضعیت‌ها و گذرهای میان آن‌ها تشکیل شده است[1,2,3].

در ابتدا به منظور انجام عملیات وارسی مدل، باید سیستم مورد نظر را به طور دقیق و مبتنی بر روش‌های فرمال مدل‌سازی، مدل کرد. یکی از روش‌های فرمال به منظور مدل‌سازی، سیستم تبدیل گراف[6] می‌باشد. سیستم تبدیل گراف به عنوان یک زبان فرمال گرافیکی محسوب می‌شود که

1 - Model checking
2 - Property
3 - Model checker
4 - Temporal Logic
5 - State space
6 - Graph transformation system

تشکیل شده است از یک گراف به عنوان گراف میزبان[1] و تعدادی قوانین تولید[2] که بر روی آن گراف میزبان اولیه اعمال می‌شود. قوانین موجود در سیستم تبدیل گراف، رفتارهای سیستم و چگونگی تغییر و تحول سیستم را به طور گرافیکی مشخص می‌کنند. به همین خاطر سیستم تبدیل گراف به خوبی قادر به مدل کردن ساختار پویای سیستم می‌باشد. همچنین، گراف ها ابزار بسیار سودمندی برای توصیف ساختارها و سیستم های پیچیده، به طوری که به صورت مستقیم و شهودی قابل درک باشند، هستند[4,5,6].

از طرف دیگر، نیازمندی‌ها و خصوصیات مورد نظری که باید در سیستم موجود باشد، می‌توانند یا به صورت فُرمال یا به صورت غیر فُرمال (روش‌هایی که بر مبنای تئوری ریاضی دقیقی نمی‌باشند) بیان شوند. یک راه غیر فُرمال به منظور بیان نیازمندی‌های سیستم به طور سطح بالا، استفاده از نمودار use-case می باشد. در این نوع نمودار، هر use-case نشان دهندهٔ توصیف غیر فُرمالی از مجموعهٔ سناریوهای ممکن شامل در سیستم مورد نظر می‌باشد. به هر حال از آنجایی که use-caseها ذاتاً سطح بالا و غیر فُرمال می باشند، نمی توانند به عنوان پایه و اساسی بمنظور آزمایش و بررسی های خودکار سیستم به کار بروند. از این رو بمنظور انجام عملیات تجزیه و تحلیل دقیق‌تر و همچنین توانایی استفاده از ابزارهای خودکار در این امر، نیازمندی‌ها و خصوصیاتی که use-caseها بیان می‌کنند باید به نیازمندهایی که با استفاده از یک زبان فُرمال بیان شده‌اند ترجمه شوند. یک راه بمنظور بیان دقیق تر نیازمندی‌ها و خصوصیات سیستم، استفاده از نمودارهای توالی پیغام[3] (MSC) می باشد[2, 7]. MSC در واقع به عنوان یک زبان گرافیکی شناخته می شود که قابلیت بیان نیازمندی ها و خصوصیات سیستم را در قالب سناریوها دارا می‌باشد. از ایدهٔ نمودارهای MSC در UML نیز استفاده شد که به آن نمودار توالی[4] گفته می‌شود. با این همه، باز هم نمودار MSC و نمودار توالی دارای کاستی‌هایی در بیان سناریوها و نیازمندی‌های سیستم می‌باشند. به عنوان مثال ممکن است در یک سناریو نیاز باشد که بین رفتارهای الزامی و اختیاری تفاوت قائل شد یا این که یک سناریو را به عنوان سناریوای اجباری و سناریوایی را به عنوان سناریوای اختیاری مشخص کرد و یا این که لازم به بیان این باشد در

1 - Host graph
2 - Production rules
3 - Message Sequence Chart
4 - Sequence Diagram

نقطه‌ای از سناریو روند اجرای سیستم الزاماً همچنان ادامه پیدا کند و در نقطه‌ای دیگر لازم به پیشروی سناریو نباشد و از این قبیل مسائل. به همین دلیل، در سال ۱۹۹۸ کاستی های موجود در نمودار MSC توسط Warner Damm و David Harel بیان شد و نتیجهٔ آن توسعه و گسترش MSC به منظور ایجاد یک روش غنی‌تر برای بیان نیازمندی‌ها و خصوصیات سیستم در قالب سناریوها شد. این زبان گرافیکی گسترش یافته، نمودار توالی جریان‌دار[1] (LSC) نامیده شد[8]. از آنجا که نمودار LSC از سمانتیک معنایی دقیقی بهره‌مند می‌باشد، می‌توان با استفاده از این نمودار، نیازمندی‌ها و خصوصیات سیستم را به طور دقیق و بدون ابهامی در قالب سناریوها بیان کرد.

همانطور که پیشتر گفته شد، یکی از روش‌های متداول بمنظور بررسی فرمال سیستم‌ها وارسی مدل می‌باشد. حال آنکه مشکل اصلی که عملیات وارسی مدل با آن روبرو است مشکل انفجار فضای حالت[2] می‌باشد. این مشکل در سیستم هایی که دارای فضای حالت بسیار بزرگ یا بینهایت هستند به وجود می آید به این معنی که حجم فضای حالت سیستم مورد نظر به اندازه ای بزرگ است که وارسی کنندهٔ مدل با کمبود حافظه مواجه شده و نمی تواند عملیات وارسی مدل را به پایان برساند. از اینرو طی سال‌های گذشته تا کنون کارهای زیادی در زمینهٔ مبارزه با انفجار فضای حالت صورت گرفته و راهکارهای متعددی به منظور کاهش و مقابله با این مشکل توسعه داده شده است. برخی از این راهکارها عبارت اند از : وارسی مدل نمادین[3]، کاهش ترتیب جزئی[4]، تکنیک تقارن[5] و تکنیک های on the fly [1, 3].

از طرف دیگر، در کاربردهای واقعی، بیان برخی از ویژگی‌ها و خصوصیات سیستم مورد نظر از قبیل ارتباط بین دو یا چند مولفه، یا خصوصیاتی که در آنها ترتیب توالی رخدادهای بین مولفه‌های سیستم اهمیت داشته با استفاده از فرمول‌های منطق زمانی دشوار و در مواردی غیر ممکن می‌باشد. بنابراین اخیراً روش‌هایی در زمینهٔ بررسی فرمال سیستم‌های نرم‌افزاری ارائه شده‌اند که از سناریوها به منظور بررسی خصوصیات سیستم استفاده کرده‌اند. در این روش‌ها به جای استفاده از فرمول‌های منطق زمانی برای بیان خصوصیات سیستم از سناریوهایی که در قالب نمودار LSC یا

1 - Live Sequence Chart
2 - State space explosion
3 - Symbolic model checking
4 - Partial order reduction
5 - Symmetry

نمودارهای VTS [1] بیان شده‌اند بمنظور انجام عملیات بررسی سیستم نرم‌افزاری استفاده شده است. روند کلی در این راهکارها به این صورت است که خصوصیت‌ها و ویژگی‌های مورد نظر که به صورت سناریوهای رفتاری در قالب نمودارهای تعاملی تعریف شده‌اند را در ابتدا به آتاماتا یا فرمالیسم‌های سطح پایین‌تری تبدیل کرده و سپس به همراه مدل سیستم بمنظور انجام عملیات بررسی به وارسی کنندهٔ مدل تحویل داده می‌شوند. نتیجهٔ حاصل شده از روش‌های مبتنی بر سناریو این بوده که، آن دسته از خصوصیات که با استفاده از منطق زمانی قابل بیان نمی‌باشند و نتیجتاً بررسی آنها امکان‌پذیر نبوده، را می‌توان بیان کرده و سپس آنها را بررسی کرد[7, 9, 10] .

۱ – ۲ – اهداف پایان‌نامه

همانطور که در بخش قبل بیان شد، وارسی مدل دارای مشکلی به نام انفجار فضای حالات بوده، که بررسی خصوصیات را برای سیستم‌های بزرگ امری دشوار و در برخی موارد امری نشدنی ساخته است. از طرف دیگر، در کاربردهای واقعی، برخی از خصوصیات و ویژگی‌هایی که مربوط به ارتباط بین دو یا چند مولفه و ترتیب رخدادهای سیستم می‌باشند را نمی‌توان از طریق فرمالیسم منطق زمانی و فرمول بندی منطق زمانی بیان کرد و در نتیجه نمی‌توان به راحتی عملیات بررسی را برای این دسته از خصوصیات انجام داد. از این رو در این پایان‌نامه سعی شده که در زمینهٔ سیستم‌های تبدیل گراف راهکاری خودکار و مبتنی بر سناریو بمنظور بررسی سیستم‌هایی که با استفاده از فرمالیسم سیستم تبدیل گراف مدل شده‌اند، ارائه شود. هدف از ارائهٔ این راهکار این است که بتوان با استفاده از این روش برخی از ویژگی‌ها و خصوصیاتی که بررسی آن‌ها با استفاده از روش‌های معمول وارسی مدل به دلیل مشکل انفجار فضای حالت امکان پذیر نمی‌باشد را در زمینهٔ سیستم‌های تبدیل گراف امکان‌پذیر کرد. همچنین مزیت دیگری که این روش مبتنی بر سناریو خواهد داشت، سهولت در بیان خصوصیات و بررسی آن دسته از ویژگی‌هایی است که با استفاده از منطق زمانی به دشواری بیان می‌شوند.

1 - Visual Timed event Scenarios

۱ – ۳ – ساختار پایان‌نامه

ساختار پایان نامه به این صورت می‌باشد که: در فصل دوم مفاهیم مربوط به فرمالیسم سیستم تبدیل گراف، نمودار LSC، وارسی مدل و ابزار مورد استفاده که از پیش‌نیازهای اصلی بحث می‌باشند، معرفی شده‌اند. در فصل سوم مروری بر روش‌ها و کارهای انجام شده در زمینهٔ وارسی مدل سیستم‌های تبدیل گراف، روش‌های مقابله با انفجار فضای حالت در امر وارسی مدل سیستم‌ها و راهکارهای مبتنی بر سناریو در بررسی سیستم‌های نرم‌افزاری، صورت گرفته است. فصل چهارم به جزئیات و تعاریف فرمال مربوط به راهکار مبتنی بر سناریو و نحوهٔ ادغام و تطبیق آن با سیستم تبدیل گراف اختصاص داده شده است. در فصل پنجم، چهار نمونهٔ عملی با استفاده از روش مبتنی بر سناریو مورد بررسی قرار گرفته‌اند و در آخر، در فصل ششم جمع‌بندی و نتیجه‌گیری از بحث ارائه شده است.

فصل دوم

مروری بر ادبیات موضوع

در این فصل مفاهیم پایه‌ایی که مرتبط با موضوع کار پایان‌نامه می‌باشند، معرفی می‌شوند. در ابتدا سیستم تبدیل گراف مورد بررسی قرار می‌گیرد، در ادامه نقش سناریوها در سیستم‌های نرم‌افزاری و همچنین نمودار LSC و اجزاء آن تعریف می‌شوند، پس از آن عملیات وارسی مدل با جزئیات بیشتر معرفی خواهد شد و در آخر نیز ابزار Groove به طور خلاصه معرفی شده است.

۲ – ۱ – سیستم تبدیل گراف[1]

گراف‌ها ابزاری بسیار سودمند بمنظور توصیف ساختارها و سیستم‌های پیچیده و همچنین ابزار مناسبی بمنظور مدل کردن ایـــده‌ها و مـفاهـــیم، طوری که به صورت مستقیم و شهودی قابل درک باشند، به حساب می‌آیند. علاوه بر ساختار، با استفاده از گراف‌ها می‌توان رفتار سیستم‌ها را نیز توصیف کرد[5].

1 - Graph Transformation System

معمولاً تعریفی که از مدل ارائه می‌شود عبارت است از یک توصیف انتزاعی از سیستم بدون در نظر گرفتن جزئیات و مسائل مربوط به پیاده‌سازی. با استفاده از مدل سیستم‌ها می‌توان عملیات تجزیه و تحلیل و بررسی الگوریتم‌های به کار رفته در آن سیستم‌ها را با سهولت بیشتری انجام داد[4]. حـال اگر ساختار مدل‌ها با استفاده از گراف توصیف شوند، می توان از قوانین تبدیلات گراف نیز به مـنظور مـشخص کردن چگونگی تغییر و تحول آن مدل‌ها در طی روند اجرای سیستم بهره گـرفت[5].

گرامر گراف‌ها و سیستم تبدیل گراف در اواخر دهۀ ۶۰ توسعه پیدا کردند و پس از آن در زمینه‌هایی مانند تشخیص الگو، طراحی ساختار کامپایلرها، تعیین مشخصات سیستم‌های نرم‌افزاری به طور فرمال، طراحی شمای VLSI، طراحی سیستم بانک های اطلاعاتی، مدل‌سازی سیستم‌های همروند، زبان‌های برنامه‌نویسی گرافیکی و بسیاری از کاربردهای دیگر بطور موثر مورد استفاده قرار گرفته‌اند.

کاربست پذیری گستردۀ گرامرهای گراف به این دلیل می باشد که گراف‌ها به طور گرافیکی ذاتاً روشی ساده و شهودی را به منظور توصیف وضعیت های پیچیده فرآهم می آورند. علاوه بر گرامر گراف‌ها، با استفاده از تبدیلات بین گراف می‌توان چگونگی تغییر و تحول مدل سیستم و رفتارهای پویای سیستم را توصیف کرد. از این رو گرامر گراف‌ها به همراه تبدیلات گراف را می توان به عنوان یک زبان برنامه نویسی گرافیکی در حوزۀ مهندسی نرم‌افزار در نظر گرفت[11].

سیـستم تبدیل گراف و گرامر گراف یـک زبان برنامه نویسی گرافیکی را تشکـیل‌مـی‌دهد به طوری که مدل داده‌ای آن بر مبنای گراف می‌باشد. این فُرمالیسم از قوانین تبدیل گراف به منظور مدیریت و دستکاری گراف ها استفاده می کند که اعمال هر قانون به یک گراف، منجر به تولید یک گراف جدید می‌شود[12]. در واقع سیستم تبدیل گراف در راستای تعمیم و گسترش راهکارهای قدیمی در سیستم های بازنویسی مانند نظریۀ زبان‌ها و گرامرهای چامسکی یا سیستم بازنویسی عبارات[1] رشد و توسعه پیدا کرد[5]. در ادامۀ این بخش به معرفی مفاهیم مهم و کاربردی که در گرامر گراف‌ها و سیستم‌های تبدیل موجود می‌باشد پرداخته شده است.

1 - Term rewriting