علوم و مهندسی کامپیوتر
نشریه علمی پژوهشی انجمن کامپیوتر ایران
مجلد ۳، شماره ۳ (الف)، پائیز ۱۳۸۴
صفحات ۴۳-۵۳
مقاله عادی

الگوریتمی کارا در وارسی الگو برای زیرنظریه‌ای از حساب Mu ×

محمد ایزدی[۳،۲،۱]   علی موقر رحیم آبادی[۲،۱]

[۱] دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران
[۲] پژوهشکده علوم کامپیوتر، پژوهشگاه دانشهای بنیادی، تهران، ایران
[۳]پژوهشگاه علوم انسانی و مطالعات فرهنگی ، تهران، ایران

چکیده

در این مقاله ضمن توصیف کلی مسئله درستی‌یابی ویژگی‌های سیستم‌ها و مساله وارسی الگو، منطق زمانی نقطه ثابت یا حساب Mu و نظریه عمومی خودکارهای بازگشتی مرتبه اول و همچنین روش‌های موجود در وارسی الگو در حساب Mu ارائه شده است. الگوریتمی جدید برای وارسی الگو وقتی ویژگی‌های سیستم در زیرنظریه‌ای از حساب Mu توصیف شوند با استفاده از ترجمه فرمول‌های حساب Mu به خودکارها ارایه می‌شود. مهمترین مزیت این الگوریتم نسبت به سایر الگوریتم‌های ارایه شده تاکنون آن است که پیچیدگی زمانی آن نسبت به اندازه مدل سیستم خطی است و قدرت توصیف زیرنظریه در نظر گرفته شده از سایر زیرنظریه‌هایی که دیگران در نظر گرفته‌اند بیشتر است.

کلمات کلیدی: وارسی الگو، درستی‌یابی، حساب Mu ، منطق‌های زمانی نقطه ثابت، نظریه خودکارها

۱- مقدمه

به موازات توسعه و پیشرفت طراحی و تولید سیستم‌های نرم‌افزاری و سخت افزاری، مسئله اعتبارسنجی طراحی ارائه شده برای چنین سیستمی به ویژه برای سیستم‌های واکنشی و همروند به دلایل متعددی از اهمیت خاصی برخوردار شده است. مسئله اعتبارسنجی یک سیستم عبارت است از ارائه روشی برای تضمین آن که طراحی ارائه شده کلیه ویژگی‌های لازم سیستم را ارضا می‌کند. از جمله دلایل اهمیت این مسئله می‌توان به اهمیت اقتصادی کشف نواقص یا خطاهای سیستم قبل از تولید انبوه آن و اهمیت امنیت عملکردی سیستم‌های ویژه مانند سیستم کنترل ترافیک هوایی اشاره کرد. وارسی الگو روشی برای اعتبارسنجی سیستم‌های همروند یا واکنشی دارای تعداد حالات متناهی است که مزیت اصلی آن در امکان انجام کاملاً خودکار آن می‌باشد. فرایند وارسی الگو در واقع جستجوی کل فضای حالات سیستم برای یافتن پاسخ این پرسش است که آیا ویژگی‌های خاصی در یک یا چند حالت مشخص سیستم و در موارد دیگری در تمامی حالات سیستم ارضا می‌شوند یا نه؟ وارسی الگو شامل سه مرحله اصلی است [۱] : مرحله مدل سازی سیستم که در آن طراحی ارائه شده به فرمالیزمی قابل قبول از دیدگاه ابزارهای وارسی الگو تبدیل می‌شود، مرحله توصیف که در آن ویژگی‌های لازم سیستم در یک زبان منطقی دقیق بیان می‌شوند و مرحله درستی‌یابی که در آن بر اساس الگوریتمی قابل اجرای خودکار، ارضا شدن ویژگی‌های توصیف شده در مرحله دوم توسط مدل سیستم ساخته شده در مرحله اول مورد بررسی قرار گرفته و پاسخ بله/خیر تولید می شود.این فرایند در صورت منفی بودن پاسخ مرحله درستی‌یابی می‌تواند توسط مرحله تکمیلی عیب یابی ادامه یابد.

برای مرحله مدل‌سازی معمولا از ساختارهای کریپکی یا خودکارها استفاده می‌شود. یک سیستم دارای تعداد حالات ممکن متناهی را می‌توان توسط یک گراف گذار مدل کرد. نوع خاصی از گراف‌های گذار که در آنها هر رأس یا هر حالت سیستم به کلیه گزاره‌های یک مجموعه مشخص AP ارزش‌دهی می‌کند (درستی یا نادرستی منتسب می‌کند) را ساختار کریپکی می نامند. علی الاصول یک سیستم با حالت‌های متناهی را توسط یک مدل کریپکی مدل‌سازی می‌کنیم. برای مرحله توصیف ویژگی‌های لازم سیستم روشهای صوری وجود دارند که یکی از موفق‌ترین آنها از نظر قدرت بیان، توصیف ویژگی‌های سیستم در زبان

منطق‌های زمانی است. عملگرهای زمانی پایه مانند «گاهی» (F)، «همیشه» (G) و «بعداً» (X) اگر به زبان منطق گزاره ای استاندارد اضافه شوند می‌توانند بسیاری از خصوصیات صحت عملکرد یک سیستم (به خصوص سیستم‌های واکنشی) را بیان کنند. به عنوان مثال فرمول بسته $G\ (send \Rightarrow F\ receive)$ بیان می‌کند که هرگاه یک پیام ارسال شود، بالاخره دریافت خواهد شد.

اگر سیستمی توسط یک گراف گذار M مدل شود و ویژگی‌های لازم آن توسط فرمول‌هایی از منطق زمانی خاصی بیان گردند در این صورت مسئله وارسی الگو را به شکل زیر بیان می‌کنیم:

مسئله وارسی الگو: گراف گذار با تعداد حالت متناهی M که حالت شروع آن s_0 است و فرمول f توصیف شده در منطق زمانی T مفروض هستند. آیا $M,s_0 \models f$ ؟ به عبارت دیگر آیا مدل M، در حالت s_0 فرمول f را ارضاء می‌کند؟

در حل مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود مجموعه‌ای از الگوریتم‌ها ارائه شده‌اند که هم از دیدگاه نظری و هم از دیدگاه پیاده‌سازی مورد بررسی و تحلیل قرار گرفته‌اند. از دیدگاه تاریخی به کارگیری این سیستم منطقی در توصیف ویژگی‌های سیستم‌های کامپیوتری از آغاز دهه ۱۹۸۰ و با به کارگیری آن در توصیف ویژگی‌های برنامه‌های همروند در [۳،۲] آغاز شد. نشان داده شده است که این سیستم دارای قدرت بیان بالاتری از سایر منطق‌های زمانی شامل LTL، CTL و *CTL است و تمامی فرمول‌های این سیستم‌های منطقی به آن قابل ترجمه‌اند [۴] و همچنین ثابت شده است که قدرت بیان حساب Mu معادل خودکارهای بر درخت‌های نامتناهی است [۱۳،۶،۵].

اگر بخواهیم از دیدگاه نظریه پیچیدگی محاسباتی به مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود بپردازیم در درجه اول بایستی به این نکته دقت کنیم که عوامل موثر که بایستی در تحلیل پیچیدگی هر الگوریتم حل این مساله در نظر گرفته شوند عبارتند از ۱) اندازه مدل سیستم و ۲) اندازه و عمق فرمول توصیف کننده ویژگی لازم آن. اما از میان این دو عامل اولی از اهمیت بیشتری برخوردار است. چنان که پنولی و دیگران استدلال کرده اند عامل اول یعنی اندازه مدل سیستم در عمل اغلب بسیار بزرگ است در حالی که اندازه و عمق فرمول توصیف کننده ویژگی لازم آن سیستم در مقایسه با اندازه مدل بسیار کوچک‌اند [۷]. بنابراین علاقه‌مندی اصلی محققین این حوزه الگوریتم‌هایی است که هر چه بیشتر نسبت به اندازه مدل کارا باشند. الگوریتم‌های معمول که مبتنی بر استفاده از روش تکرار یا جستجوی تمامی فضای حالات مساله ارایه شده‌اند دارای زمان اجرای از مرتبه $O((|M||f|)^{k+1})$ هستند که در آنها |M| اندازه مدل، |f| طول رشته f و k عمق فرمول f بر اساس عملگرهای نقطه ثابت است. یعنی زمان اجرای این گونه الگوریتم‌ها از مرتبه چند جمله‌ای نسبت به اندازه مدل و از مرتبه نمایی نسبت به عمق فرمول توصیف‌کننده هستند. ثابت شده است که وارسی الگو برای حساب Mu در حالت کلی نسبت به عمق فرمول توصیف کننده ویژگی، مساله‌ای در $NP \cap coNP$ است [۸].

با توجه به آن که بنابر قضیه فوق یافتن یک الگوریتم با زمان چند جمله‌ای به نسبت عمق فرمول یک مساله باز و دشوار است و همچنین با توجه به اهمیت عامل اول یعنی اندازه مدل علاقه‌مندیم زمان اجرای الگوریتم وارسی الگو در حساب Mu نسبت به اندازه مدل *خطی* باشد هر چند نسبت به عمق فرمول توصیف کننده ویژگی نمایی باقی بماند. اگرچه ایده آل، خطی بودن الگوریتم نسبت به اندازه مدل و چندجمله‌ای نسبت به عمق فرمول است.

با توجه به پیچیدگی حل مساله وارسی الگو در حساب Mu رویکرد محققین این موضوع بیشتر به زیرکلاسهایی از فرمول‌های این سیستم منطقی است. الگوریتم‌هایی مانند الگوریتم طبیعی که دربخش ۴ این مقاله آن را مرور می‌کنیم و مساله را در حالت کلی حل می‌کنند یا الگوریتم‌هایی مانند الگوریتم پایه و دولوکس به دلیل نوع محدودیتی که بر فرمول‌ها اعمال می‌کنند همگی دارای پیچیدگی زمانی چند جمله ای (غیر خطی) بر اساس اندازه مدل سیستم و پیچیدگی زمانی نمایی بر اساس عمق فرمول توصیف کننده ویژگی هستند. به کارگیری محدودیتهای مناسب و استفاده از روشهای تلفیقی مانند روش مشخص‌سازی منطق‌های زمانی نقطه ثابت توسط خودکارها این امید را ایجاد کرده است که مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود با کمترین اعمال محدودیت دارای الگوریتمی با زمان خطی بر اساس اندازه مدل است. قبل از این مقاله دو زیرسیستم حساب Mu توسط امرسون و دیگران در [۱۰،۹،۸] و زیرسیستم بسیار محدودی در [۱۱] مورد بررسی قرار گرفته‌اند که وارسی الگو برای آنها دارای زمان خطی بر اندازه مدل است. ما در این مقاله الگوریتمی برای وارسی الگو بر اساس توصیف ویژگی‌ها در زیرنظریه دیگری از حساب Mu و با استفاده از نظریه خودکارها ارایه می‌کنیم که مرتبه پیچیدگی زمانی آن $O(|M|.|f|^{k+1})$ است و این مرتبه پیچیدگی زمانی نسبت به اندازه مدل خطی است. نشان خواهیم داد که اگرچه دو زیرسیستم ارایه شده توسط امرسون و دیگران در [۱۰،۹،۸] و همچنین زیرسیستم ارایه شده در [۱۱] نیز دارای زمان اجرای خطی بر اندازه مدل هستند اما از دیدگاه قدرت بیان زیرسیستمی که ما درنظر گرفته‌ایم قوی‌تر و دارای محدودیت نحوی کمتری است. شمای کلی از روشی که در این مقاله به کار گرفته شده است و الگوریتم حاصل قبلا توسط نگارنده در مقاله کنفرانس‌های [۲۰،۱۹،۱۸] ارایه شده‌اند.

ساختار این مقاله به شکل زیر است. در بخش دوم گرامر و معناشناسی حساب Mu در حالت کلی و در بخش سوم گرامر و معناشناسی قرایت خاصی از حساب Mu با مدل زمانی درختی که به حساب mKn موسوم است را ارایه می کنیم. در بخش چهارم مروری بر روشهای موجود در وارسی الگو در حساب Mu خواهیم داشت. بخش پنجم به معرفی کلی روش مشخص‌سازی فرمول‌های منطق‌های زمانی نقطه ثابت توسط خودکارها را معرفی می‌کند. بخش ششم به مرور نظریه عمومی خودکارها با تاکید بر خودکارهای .بازگشتی مرتبه اول اختصاص دارد. بخش هفتم به قضیه اساسی تناظر میان خودکارهای بازگشتی مرتبه اول و فرمول‌های حساب Mu می‌پردازد و در نهایت در بخش هشتم الگوریتمی برای وارسی الگو وقتی که ویژگی‌های لازم سیستم به زبان حساب Mu بیان شوند و مدل سیستم یک خودکار متعارف بوچی یا یک ساختار کریپکی باشد ارایه شده و پیچیدگی آن بررسی می گردد. در این بخش نشان خواهیم داد که الگوریتم پیشنهاد شده برای زیرکلاسی از فرمول‌های حساب Mu که معرفی خواهند شد بر اساس اندازه مدل خطی است. بخش پایانی به جمع‌بندی و نتیجه‌گیری می‌پردازد.

۲- حساب Mu تعبیر شده بر مدلهای کریپکی

حساب Mu گزاره ای مبتنی بر اضافه کردن دو عملگر نقطه ثابت شامل عملگر کوچکترین نقطه ثابت (μ) و عملگر بزرگترین نقطه ثابت (ν) به زبان منطق گزاره ای است. به طور شهوری، در حساب Mu، موجهات زبان به وسیله الگوهای درختی که به طور بازگشتی تعریف می شوند مشخص می گردند. به عنوان مثال فرض کنید می‌خواهیم این عبارت «در تمامی مسیرهای محاسبه ممکن گزاره p بالاخره درست خواهد بود.» را در یک منطق زمانی بیان کنیم. در نظام منطقی CTL این عبارت به صورت AF p بیان می‌شود که در آن A سور عمومی بر تمامی مسیرهای محاسبه ممکن و F عملگر زمانی بالاخره است. این عبارت را در حساب Mu توسط فرمول $mZ.p \vee AX\ Z$ نشان می‌دهیم. معنای این فرمول کوچکترین نقطه ثابت تابع $p \vee AX\ Z$ است که در آن Z یک متغیر گزاره‌ای اتمی و AX عملگر بعداً در تمامی مسیرها است. به عبارت دیگر اگر به ساختار مدل درخت محاسبات ممکن یک سیستم به شکل بازگشتی نگاه کنیم فرمول AF p از منطق CTL می‌تواند به صورت کوچکترین مجموعه از راسهای این درخت بیان شود به نحوی که p در ریشه درخت یا AX p حداقل در یکی از زیردرختهای ریشه راست است.

نحو و معناشناسی حساب Mu را به دو روش می‌توان ارایه داد: یکی با استفاده از ساختارهای کریپکی در تعبیر فرمول‌ها که از دیدگاه منطقی روشی مرسوم‌تر است و دیگری استفاده از درخت‌های نامتناهی با تعداد شاخه یا انشعاب متناهی در تعبیر فرمول‌ها که اگرچه معادل روش قبلی است اما برای هدف ما یعنی ترجمه فرمول‌ها به خودکارها مناسب‌تر است. در ادامه این بخش به تعاریف صوری و سپس معناشناسی حساب Mu با استفاده از ساختارهای کریپکی در تعبیر فرمول‌ها می پردازیم و روش دوم را در بخش بعدی مقاله ارایه می‌کنیم.

۲-۱ نحو یا گرامر حساب Mu

مجموعه فرمول‌های زبان حساب Mu گزاره ای (زبان L_m) فرمول‌هایی هستند که توسط قواعد ۱ تا ۶ زیر تولید می شوند:

۱- ثوابت گزاره ای اتمی P، Q، …

۲- متغیرهای گزاره ای اتمی Y، Z، …

۳- EX p وقتی p یک فرمول L_m باشد.

۴- ~ p وقتی p یک فرمول L_m باشد.

۵- p∧q وقتی p , q فرمول‌های از L_m باشند.

۶- $mY.p(Y)$ که p(Y) فرمولی است که بر متغیر اتمی Y به طور نحوی یکنوا است. (یک فرمول به طور نحوی بر متغیر Y یکنواست اگر هر وقوع Y در آن در محدوده تعدادی زوج از علائم نفی (~) قرار گیرد.)

سایر اتصالات منطقی به طور متعارف بر اساس اتصالات تعریف شده در بالا، درنظر گرفته می شوند. مثلاً p∨q اختصاری برای (((~p) ∧ (~q))~) خواهد بود. به طور خاص دقت کنید که AX p اختصاری برای EX ~p~ و همچنین v(Y).p(Y) اختصاری برای $m(Y).\sim r(\sim Y)$ خواهند بود. از دیدگاه نحوی عملگرهای نقطه ثابت n, m بسیار شبیه سورهای عمومی و وجودی هستند. هر وقوع متغیر Y در زیر فرمول $mY.p(Y)$ را محدوده شده می نامیم. در غیر اینصورت Y را متغیر آزاد می گوییم. یک جمله یا فرمول بسته، فرمولی است که متغیر آزاد ندارد یعنی هر متغیر آن محدود شده به یک عملگر نقطه ثابت است. یک فرمول را به شکل نرمال مثبت (PNF) می گوییم اگر هیچ متغیری دوبار مسور نشده باشد و تمامی علائم نفی فقط بر گزاره های اتمی به کار رفته باشند. دقت کنید که با استفاده از قانون دمورگان و خصوصیت دوگانی عملگرهای n, m که در بالا به آن اشاره کردیم هر فرومول می‌تواند به شکل PNF در آید و این عمل حداکثر طول رشته را دو برابر می‌کند. سایر اصطلاحات منطقی مانند اصطلاح زیر فرمول، زیر فرمول سره، زیر جمله و زیر جمله سره و … کاملاً شبیه سایر سیستم‌های منطقی تعریف می شوند.

۲-۲ معناشناسی حساب Mu

فرض کنید Σ مجموعه‌ای از ثوابت گزاره ای اتمی و T مجموعه‌ای از متغیرهای گزاره ای اتمی و AP اجتماع این دو مجموعه باشد. (AP را مجموعه گزاره‌ها می‌نامیم.) حوزه تعبیر یا الگوی زیر که به مدل کریپکی موسوم است را جهت تعبیر کردن جملات زبان حساب Mu در آن تعریف می‌کنیم:

یک مدل کریپکی سه تایی M = (S, R, L) است که در آن:

S: مجموعه‌ای از حالات است.

R: یک رابطه دوتایی کامل است. یعنی

$$\forall s\in S\ \exists t\in S\ (s,t)\in R,\ R\subseteq S\times S$$

L: تابع کامل از نوع $S\rightarrow 2^{AP}$ است که به هر حالت از مجموعه S، مجموعه‌ای ازگزاره ها نسبت می دهد که در آن حالت درست هستند. L را تابع برچسب دهی می نامیم.

M را می‌توان به عنوان یک گراف سودار در نظر گرفت که S مجموعه رئوس آن و R مجموعه یالها و L مشخص‌کننده برچسب هر رأس است. اندازه مدل M مساوی |M|=|S|+|R| است. مجموعه توانی مجموعه S یعنی P(S) همراه با عملگرهای زیر مجموعه، اجتماع و اشتراک و کوچکترین عضو Ø و بزرگترین عضو S تشکیل یک مشبکه کامل می دهد. به طور شهوری اگر هر محمول یا هر فرمول بسته را توسط مجموعه‌ای از حالت‌ها که آن محمول در آنها درست است مشخص کنیم در این صورت مجموعه Ø یعنی کوچکترین عضو مشبکه مشخص کننده false و مجموعه S یعنی بزرگترین عضو مشبکه مشخص کننده true خواهند بود. به همین ترتیب عملگر منطقی استلزام یعنی $\forall s\in S\,[P(s)\Rightarrow Q(s)]$ که در این مشبکه معادل عملگر مجموعه‌ای زیر مجموعه بودن $(P\subseteq Q)$ است تشکیل یک ترتیب جزیی بر این مشبکه می دهد.

فرض کنید $t: 2^S\rightarrow 2^S$ تابعی بر مشبکه فوق باشد. τ را تابعی یکنوا می نامیم اگر وقتی $P\subseteq Q$ آن گاه $t(P)\subseteq t(Q)$ باشد. مجموعه P' را یک نقطه ثابت تابع t می نامیم اگر $P'=t(P')$. همچنین P' کوچکترین نقطه ثابت τ است اگر اولاً یک نقطه ثابت باشد و ثانیاً برای هر نقطه ثابت دیگر P'' داشته باشیم $P'\subseteq P''$. دقت کنید که کوچکترین نقطه ثابت هر تابع در صورت وجود منحصر به فرد است. بزرگترین نقطه ثابت نیز مشابه بالا تعریف می گردد. برای نمایش کوچکترین نقطه ثابت تابع t از $mY.t(Y)$ و برای بزرگترین نقطه ثابت از $nY.t(Y)$ استفاده می کنیم. قضیه بنیادی زیر وجود این دو نقطه ثابت را در مورد مشبکه فوق تضمین می‌کند:

قضیه تارسکی - ناستر [۱،۱۰]: فرض کنید $t: 2^s\rightarrow 2^s$ یک تابع یکنوا باشد، در این صورت:

الف) $mY.t(Y)=\bigcap\{Y: t(Y)=Y\}=\bigcap\{Y: t(Y)\subseteq Y\}$

ب) $nY.t(Y)=\bigcup\{Y: t(Y)=Y\}=\bigcup\{Y: t(Y)\supseteq Y\}$

ج) $mY.t(Y)=\bigcup_i t^i(false)$

د) $nY.t(Y)=\bigcap_i z^i(true)$

در هر دو مورد (ج) و (د) i محدود به تمام اوردینالهای حداکثر تا اندازه مجموعه S است. بنابراین اگر S مجموعه‌ای متناهی باشد i بین صفر تا |S| است و بنابراین برای مجموعه های متناهی S مقدار $mY.t(Y)$ اجتماع رشته مجموعه های در حال بزرگ شدن زیر است:

$$false\subseteq t(false)\subseteq t^2(false)\subseteq \ldots$$

و به همین ترتیب برای مجموعه های متناهی S مقدار $nY.t(Y)$ اشتراک مجموعه های در حال کوچک زیر است:

$$true\supseteq t(true)\supseteq t^2(true)\supseteq \ldots$$

هر فرمول P حاوی متغیرهای آزاد Y_1 , … , Y_n به عنوان یک نگاشت P^M از مجموعه $(2^S)^n$ به 2^s تعبیر می شود. اگر $P(Y_1, \ldots, Y_n)$ نمایش فرمول P باشد یک ارزش دهی V که با $(V_1, \ldots, V_n)$ مشخص می شود، انتساب زیر مجموعه های $V_1 \ldots V_n$ از مجموعه S به متغیرهای $Y_1 \ldots Y_n$ به ترتیب است. مقدار فرمول P در اثر ارزش دهی V را با $P^M(V)$ نشان می‌دهیم. P^M یک عملگر یا یک تبدیل کننده محمولات است که به طریقه استقرایی زیر تعریف می شود:

۱) به ازای هر ثابت گزاره ای اتمی $P\in AP$ داریم:

$$P^M(V)=\{s: s\in S, P\in L(s)\}$$

۲) $Y_i^M(V)=V_i$

۳) $(p\wedge q)^M(V)=p^M(V)\cap q^M(V)$

۴) $(\sim P)^M(V)=S\setminus(P^M(V))$

۵) $(EX\ P)^M(V)=\{s\mid \exists E\in P^M(V), (s,t)\in R\}$

۶) $(mY.P(Y)^M)(V)=\bigcap\{S'\subseteq S\mid P(Y)^M(S', V_2, \ldots, V_n)\subseteq S'\}$

دقت کنید که محدودیت به طور نحوی یکنوا بودن فرمول‌ها که در بخش قبل بر فرمول‌های زبان حساب m تحمیل کردیم شرط لازم امکان تعریف و تعیین کوچکترین (و بزرگترین) نقطه ثابت طبق بند ۶ بالا است.

به جز حساب Mu منطق‌های زمانی دیگری نیز وجودارند که از نظر قدرت بیان با یکدیگر متفاوتند. سیستم منطق زمانی خطی گزاره ای یا LTL که به منطق زمانی خطی استاندارد نیز موسوم است، همچنین منطق محاسبات درختی یا CTL و توسعه آن *CTL مهمترین آنها هستند [۱] . این ویژگی مهم حساب Mu که قدرت بیانی آن از منطق‌های زمانی معرفی شده بیشتر است [۳] و امکان ترجمه سیستماتیک همه آنها به این حساب وجود دارد [۴] از این جهت مهم است که اولاً حل مسئله وارسی الگو برای آن را حلی برای تمامی منطق‌های دیگر می‌کند و ثانیاً چون از دیدگاه شهودی فهم فرمول‌های منطق‌های CTL و CTL^* بسیار ساده تر از حساب Mu است این امکان ایجاد می شود که ویژگی‌های لازم را در یکی از آن منطق‌ها بیان کنیم اما از الگوریتم‌های وارسی الگو در حساب Mu بهره ببریم.

۳- حساب Mu موجه اندیس‌دار

در این بخش به تعاریف صوری و سپس معناشناسی حساب Mu با استفاده از از درخت‌های نامتناهی با تعداد شاخه یا انشعاب متناهی در تعبیر فرمول‌ها می‌پردازیم. چنین روشی را اصطلاحا حساب Mu موجه اندیس‌دار می‌نامند و سیستمی که ما در این بخش به این روش ارایه می‌کنیم را همچون [۱۲] سیستم mKn نامیده‌ایم.

تعریف ۳-۱ یک مدل انشعابی M یک درخت کامل نامتناهی است که هر رأس آن توسط مجموعه‌ای از گزاره ها (زیرمجموعه‌ای از Z) بر چسب زده شده است: $M: N^* \to P(Z)$. تابع بر چسب گذاری فوق یک تابع جزیی است یعنی برخی رئوس درخت می‌توانند برچسب نداشته باشند. مدل انشعابی M را n–انشعابی می نامیم اگر و فقط اگر M یک درخت کامل n– انشعابی باشد. در این صورت مجموعه تمامی حالات مدل M را با St(M) نشان می‌دهیم.

نحو یا گرامر سیستم mKn: عدد ثابت $n \in N$ را در نظر بگیرید. مجموعه فرمول‌های حساب Mu موجه اندیس دار mKn را توسط گرامر انتزاعی زیر تعریف می کنیم:
$$f ::= z \mid \neg f \mid f_1 \wedge f_2 \mid X_i f \mid mz.f$$
که در آن z متغیری بر مجموعه Z و i بر [n] (مجموعه اعداد طبیعی از یک تا n) است. مانند حساب Mu خطی هر وقوع z در فرمول f بایستی مثبت باشد.

معناشناسی یا سمانتیک سیستم mKn: فرض کنید n یک عدد طبیعی و M یک مدل n–انشعابی باشد. مجموعه حالاتی از M که فرمول f از حساب mKn را ارضاء می کنند با $\|f\|_M$ نشان می دهیم و آن را توسط قواعد بازگشتی زیر تعریف می کنیم:
$$\|Z\|_m = \{S \in St(M) \mid z \in M_{(s)}\}$$
$$\|\neg f\|_m = St(M) \setminus \|f\|_m$$
$$\|f \wedge f'\|_m = \|f\|_m \cap \|f'\|_m$$
$$\|X_i f\|_M = \{s \in St(M) \mid s.i \in \|f\|_M\}$$
$$\|mz.f\|_m = \cap\{W \subseteq St(M) \mid \|f\|_{M[W/Z]} \subseteq W\}$$
که در آن M[W/Z] به روش زیر تعریف می شود:
$$M[W/Z]_{(S)} = \begin{cases} M_{(S)} \cup \{z\} & S \in W \\ M_{(S)} \setminus \{z\} & S \in St(M) \setminus W \end{cases}$$

تعریف ۳-۲ می‌گوییم فرمول f در حالت s از مدل M راست است و می نویسیم $M,s \models f$ اگر و فقط اگر $S \in \|f\|_M$. می گوییم ϕ ابتدائاً در M راست است و می نویسیم $M \models f$ اگر و فقط اگر $M,0 \models f$. فرمول ϕ را در مدل M معتبر می نامیم اگر برای هر حالت S از مدل M داشته باشیم: $M,s \models f$. فرمول f را به طور جهانی معتبر می نامیم اگر برای هر مدل M و هر حالت s از مدل M داشته باشیم $M,s \models f$. فرمول f را ارضا شدنی می‌نامیم اگر مدل M و حالت S از مدل M یافت شوند به گونه ای که $M,s \models f$. زبان مشخص شده توسط فرمول f که با $L(f)$ نمایش داده می‌شود مجموعه تمامی مدل‌هایی است که این فرمول در آنها راست است یعنی: $L(f) = \{M \mid M \models f\}$.

تعریف ۳-۳ یک فرمول μKn را به **شکل نرمال مثبت** (PNF) می‌گوییم اگر هیچ متغیری دوبار مسور نشده باشد و تمامی علائم نفی فقط بر گزاره های اتمی به کار رفته باشند.

رده خاصی از فرمول‌های حساب mkn هستند که در ارتباط با نظریه خودکارها از اهمیت خاصی برخوردارند. در ادامه این بخش این رده از فرمول‌های حساب mkn را معرفی می کنیم.

تعریف ۳-۴ اگر ϕ یک mkn فرمول به شکل PNF باشد آنگاه ϕ را غیر عطفی می نامیم اگر و فقط اگر برای تمامی زیرفرمول‌های ϕ که به شکل $f_1 \wedge f_2 \wedge \dots \wedge f_{m-1}$ هستند داشته باشیم:

۱- برای هر $i \in [m]$ حداقل یکی از شرایط زیر برقرار باشد:
- f_i یک فرمول اتمی است که در محدوده هیچ عملگر نقطه ثابت μ نیست.
- f_i فرمولی به شکل $X_k y$ است.

۲- برای هر $i, j \in [m]$ که $i \neq j$ است اگر $f_i = X_k y$ و $f_j = X_l g$ آنگاه $k \neq l$.

تعریف ۳-۵ اگر ϕ یک mkn فرمول به شکل PNF باشد آنگاه ϕ را محافظت شده می نامیم اگر و فقط اگر هر زیرفرمول نقطه ثابت $sz.y$ از آن فرمول مستقیما در محدوده یک عملگر X_k قرار داشته باشد و هر وقوع z در ψ نیز در محدوده یک عملگر X_k قرار داشته باشد.

تعریف ۳-۶ اگر ϕ یک mkn فرمول به شکل PNF باشد آنگاه ϕ را محدود شده می نامیم اگر و فقط اگر ϕ غیر عطفی و محافظت شده باشد.

۴- الگوریتم‌های موجود در وارسی الگو برای توصیفات حساب Mu

در این بخش به مرور مجموعه‌ای از الگوریتم‌های موجود وارسی الگو بر توصیفات حساب Mu می پردازیم.

۴-۱ الگوریتم‌های وارسی الگو بر اساس نمایش صریح حالات

قضیه تارسکی - ناستر یک چهارچوب عملی نسبتاً کارا برای تعیین نقطه ثابت هر فرمول حساب Mu و در نتیجه درستی‌یابی آن فرمول ارائه می‌کند. با توجه به آن که نشان داده شده است فرمول‌های سایر منطق‌های زمانی به خصوص PLTL، CTL و CTL^* قابل ترجمه به فرمول‌هایی در زبان حسابMu هستند لازم نیست توصیف اولیه حتماً به حساب Mu انجام گیرد. به خصوص با توجه به این نکته که خوانایی و قابلیت فهم فرمول‌های منطق‌های زمانی فوق الذکر از حساب Mu بیشتر است. اگر چه دقت و قدرت توصیف حساب Mu نیز از آنها فراتر است. به عنوان مثال برای محاسبه مجموعه حالاتی ازمدل M = (S,R,L) که فرمول EF P از زبان منطق CTL در آنها درست است، از ترجمه به حساب Mu به شکل $EFP = mZ.t\,(Z)$ که τ(Z) = P ∨ EX Z استفاده می کنیم سپس با محاسبه رشته مجموعه های زیر
$$t\,(false) \subseteq t^2(false) \subseteq \ldots \subseteq t^K(false)$$
به نحوی که برای $k \leq |S|$ داشته باشیم $t^k(false) = t^{k+1}(false)$ به نقطه ثابت فرمول فوق که همان آخرین مجموعه حاصل شده است دست خواهیم یافت. دیدگاهی شهودی نسبت به محاسبه فوق را می‌توان به این گونه بیان کرد: هر

$t^{i}(false)$ متناظر با مجموعه‌ای از حالات است که می‌توان از P با حداکثر فاصله i به آنها رسید. بنابراین P از حالت S دسترسی پذیر است اگر و فقط اگر P با i مرحله قابل دسترسی از حالت S باشد که در آن i حداکثر اندازه مدل M است. به عبارت دیگر P از حالت S دسترسی پذیر است اگر و فقط اگر $S \in t^{i}(false)$ برای iای که $i \leq |M|$.

این ایده می‌تواند به سادگی تبدیل به الگوریتمی برای درستی‌یابی ویژگی‌های توصیف شده در منطق‌های زمانی فوق الذکر و حتی خود حساب Mu شود. فرمول هایی با عملگرهای پایه توسط قضیه تارسکی مورد پردازش قرار می گیرد، فرمول‌های ترکیبی با استفاده از رتبه بندی عملگرها و ترکیبات بولی از فرمول‌ها با استفاده بازگشتی از دو روش فوق بررسی می شوند. به طور کلی این روش یا این گونه الگوریتم‌ها را الگوریتم‌های با روش تکرار نقطه ثابت می نامیم.

الگوریتم طبیعی. مدل M، فرمول f و ارزش دهی V مفروضند می خواهیم مجموعه $set(f) = \{s : M, s \models f(V)\}$ یعنی مجموعه تمامی حالاتی در مدل M که فرمول f تحت ارزش دهی V در آنها درست است را بیابیم. کافی است ابتدا برای تمامی ثوابت گزاره ای P و متغیرهای گزاره ای Y مقدار اولیه set آنها را به شکل زیر تعیین می کنیم:

$$set(P) = \{S : M, s \models P\}$$
$$set(Y) = V(Y)$$

سپس به روش استقرایی (بازگشتی) با استقرا بر ساختار فرمولی set(f) برای هر فرمول f محاسبه می شود [۱].

در یک پیاده‌سازی متعارف، زمان اجرای الگوریتم فوق از مرتبه $O((|M||f|)^{k+1})$ است که در آن |M| اندازه مدل، |f| طول رشته f و k عمق فرمول f بر اساس عملگرهای نقطه ثابت است [۱۰]. (تعریف عمق یک فرمول بر اساس عملگرهای نقطه ثابت n, m همان تعریف عمق یک فرمول است اگر n, m را به جای سورهای عمومی و وجودی در نظر بگیریم.)

الگوریتم پایه. در الگوریتم طبیعی که در بالا به آن اشاره کردیم از خصوصیت یکنوا بودن توابع (یا فرمول‌هایی) که الگوریتم بایستی نقطه ثابت آنها را محاسبه کند استفاده‌ای نشده است. می‌توانیم توسعه ای از قضیه تارسکی را در نظر بگیریم که این ویژگی توابع بر روی مشبکه 2^{S} استفاده می‌کند:

قضیه تعمیم یافته تارسکی – ناستر [۱۰]. فرض کنید $t : 2^{S} \rightarrow 2^{S}$ یک تابع یکنوا باشد در این صورت :

الف) $mY.t(Y) = U_i t^{i}(Y_0)$ برای هر $Y_0 \subseteq t(Y_0) \cap mY.t(Y)$

ب) $nY.t(Y) = \bigcap_i t^{i}(Y_0)$ برای هر $Y_0 \supseteq t(Y_0) \cap nY.t(Y)$

در هر دو مورد i محدود به تمام اودینالهای حداکثر تا اندازه مجموعه S است. بنابراین اگر S متناهی باشد i بین صفر تا |S| خواهد بود و در این صورت مقدار $mY.t(Y)$ اجتماع رشته افزایشی $Y_0 \subseteq t(Y_0) \subseteq t^{2}(Y)\ldots$ و مقدار $nY.z(Y)$ اشتراک رشته کاهشی $Y_0 \supseteq t(Y_0) \supseteq t^{2}(Y)\ldots$ خواهند بود.

برای هر فرمول به جای تعریف عمق (به طور مطلق) می‌توان یک ملاک **عمق جابجایی** (ad) تعریف کرد. این ملاک به طور شهودی نشان دهنده عمق عملگرهای n, m در فرمول است که بر یکدیگر موثرند به عبارت دیگر تابعی که عملگر داخلی بر آن عمل می‌کند دارای متغیرهای آزاد از نوع متغیر عملگر بیرونی است معینی ad عمق عملگرهای n, m که به شکل زیر بر یکدیگر موثرند را نشان می دهد:

$mY.f(nz.g(Y.z))$ یا $nY.f(mz.g(Y.z))$

(برای تعریف دقیق ملاک ad که برای هر فرمول همواره کمتر یا مساوی عمق آن فرمول است به [۱۰،۴] مراجعه کنید.)

به کارگیری قضیه تعمیم یافته فوق می‌تواند زمان اجرای الگوریتم طبیعی را در مواردی که عملگرهای m و n به صورت جابجایی (موثر بر یکدیگر) نیستند کاهش دهد [۱۱]. حاصل الگوریتمی خواهد بود که برای زیرنظریه حساب Mu که آن را فارغ از عملگرهای نقطه ثابت جابجایی می‌نامند و در آن هیچ دو عملکر نقطه ثابتی نبایستی بر یکدیگر به کار روند با یک پیاده‌سازی بهبود یافته دارای پیچیدگی زمانی $O(|M| . |F|^{ad})$ و پیچیدگی حافظه $O(|M| . |f|)$ خواهد بود [۱۱].

الگوریتم دولوکس. از ویژگی یکنوا بودن فرمول‌ها و توابع در حساب Mu در مورد فرمول‌های با اشکال خاص دیگر نیز می‌توان بهره گرفت. ایده اصلی آن است که در فرمول‌هایی به شکل $mY_1 nZ_1.mY_2 nZ_2 \ldots mY_1.g(Y_1, Z_1, Y_2, Z_2, \ldots Y_n)$ با عمق جابجایی ad=2n-1 متغیرهای m بایستی همگی نسبت به یکدیگر یکنوا باشند و داخلی ترین متغیر m در تمامی نمونه هایش نیز یکنواست. هر جا که این خصوصیت یکنوا بودن برقرار باشد لازم نیست در اجرای الگوریتم طبیعی محاسبه تکرار جدید نقطه ثابت از مقادیر اولیه (یعنی برای m از false و برای n از true) شروع کنیم کافی است آخرین مقادیر تکرار قبلی الگوریتم را در اجرای مجدد به جای مقادیر اولیه در نظر بگیریم [۱۴]. اگر چه الگوریتم دولوکس از دیدگاه نظری جالب است اما زمان اجرای پیاده‌سازی شده آن هنوز از مرتبه زمانی $O(|M| . |f|)^{ad}$ است [۱۴].

۴-۲ الگوریتم‌های وارسی الگو در حساب Mu بر اساس نمایش نمادین حالات

در مواردی که تعداد کل حالات سیستم بسیار زیاد است و در واقع با مسئله انفجار حالات مواجهیم نمایش صریح حالات در یک ساختار داده‌ای امکان اجرای عملی الگوریتم را تقریباً منتفی می‌کند. در این صورت روشهای نمایش نمادین حالات می‌توانند به کار گرفته شوند. الگوریتم‌های اصلی که در حوزه نمایش نمادین حالات به کار گرفته می شوند در واقع همان الگوریتم‌های با نمایش صریح حالات هستند که بر اساس روش تکرار محاسبه نقطه ثابت مجموعه حالاتی که در آنها فرمول مورد نظر درست است را به دست می آورند. تفاوت بنیادین این الگوریتم‌ها با الگوریتم‌های قبلی روش نمایش مدلهای کریپکی و مجموعه هایی از حالات که یک فرمول در آنها درست می‌باشد. علی الاصول در این الگوریتم‌ها این مجموعه های حالات توسط نمودارهای تصمیم گیری دودویی (BDD) و به ویژه BDD های مرتب (OBDD) نمایش داده می شوند [۱۵].

اگر چه BDDها از دیدگاه پیاده‌سازی در حافظه و امکان اجرای کارای الگوریتم‌های آنها به خصوص در عیب یابی مدارات منطقی بسیار مفید هستند، اما باید توجه داشت که الگوریتم‌های وارسی الگو مبتنی بر آنها در تحلیل بدترین حالت هنوز به طور غیر قابل اجتناب ناکارا هستند [۱۶]. آزمون یک گراف ساده از دیدگاه دسترسی پذیری برای مدل M اگر M توسط یک BDD، نمایش داده شود یک مسئله PSPACE-complete است [۱۶]. عدم همگرایی تحلیل بدترین حالت از دیدگاه نظری و موفقیتهای عملی استفاده از نمایشBDD به خصوص در عیب یابی سیستم‌های سخت افزاری و مدارات منطقی باعث شده است که نظریه پیچیدگی الگوریتم‌های وارسی الگو بر اساس نمایش BDD توسعه چندانی پیدا نکند [۱۰].

۵- مشخص‌سازی منطق‌های زمانی نقطه ثابت توسط خودکارها: رویکردی تازه به مساله وارسی الگو

منطق‌های حاوی عملگرهای نقطه ثابت در ارتباطی نزدیک با منطق‌های مرتبه دوم یعنی منطق‌هایی که در آنها امکان اعمال سورها بر محمولات وجود دارد قرار دارند. عملگرهای نقطه ثابت در واقع خود عملگرهایی از نوع عملگرهای مرتبه دوم هستند. به همین دلیل نشان داده شده است که از دیدگاه قدرت بیان تفاوتی میان منطق‌های نقطه ثابت و منطق‌های مرتبه دوم وجود ندارد [۱۲]. با توجه به این که نظریه خودکارهای بوچی و رابین در زمینه حل مسایل تصمیم گیری برای دو

نظریه مرتبه دوم حساب یعنی S1S و SnS ارایه شده‌اند بنابر این می‌توان از تبدیل مساله وارسی الگو به عنوان یک مساله منطقی به یک مساله در حوزه نظریه خودکارها سخن گفت. به عبارت دیگر از یک طرف منطق‌های نقطه ثابت خود در واقع منطق‌های مرتبه دوم هستند و از طرف دیگر خودکارهای بوچی و رابین و در حالت کلی‌تر نظریه خودکارهای بازگشتی فرمالیزمی برای توصیف نظریه های مرتبه دوم ارایه کرده اند بنابر این یک راه حل میانه برای مساله وارسی الگو تبدیل فرمول‌های منطق‌های نقطه ثابت به معادل‌هایی از نوع خودکارها و سپس ارایه الگوریتم وارسی الگو در ارتباط میان خودکار توصیف کننده ویژگی لازم سیستم و مدل کریپکیایی سیستم است. مزیت عمده قابل تصور برای چنین راه حلی استفاده از مجموعه تمامی نتایجی است که در حوزه نظریه خودکارها به ویژه از دیدگاه الگوریتمی و نظریه پیچیدگی تاکنون حاصل شده است.

از دیدگاه نظریه خودکارها یک خودکار می‌تواند بر اشیاء متناهی یا نامتناهی تعریف شود. اشیاء نامتناهی مورد علاقه یا از نوع رشته های نامتناهی یا درختهای نامتناهی هستند. تفاوت اصل خودکارها بر اشیاء نامتناهی شرط پذیرش آنها (این که به چه شرطی آن خودکار ورودی یا شرایط موجود را می پذیرد) است. به عنوان مثال خودکارهای متعارف بوچی و رابین را می‌توان با یک فرمالیزم تعریف کرد فقط با این تفاوت که شرط پذیرش آنها با هم متفاوت است. از طرف دیگر یک خودکار بر اشیاء نامتناهی می‌تواند از نوع خودکارهای متعارف یا خودکارهای تناوبی باشد. در یک خودکار متعارف عدم قطعیت در یک حالت را به شکل وجود هر دو گذار (q,z',q') و (q,z",q") نشان می‌دهیم که در آنها q' و q" متفاوتند و z' و z" دارای انحصار متقابل نیستند. وجود این دو گذار به این معناست که اگر خودکار در حالت q باشد و شرط z' برقرار باشد به حالت q' و اگر شرط z" برقرار باشد به حالت q" خواهد رفت. چون z' و z" دارای انحصار متقابل نیستند به طور غیر قطعی یکی از دو گذار فوق انتخاب خواهد شد. چنین انشعابی در واقع یک انشعاب فصلی است یعنی در حالت q کافی است فقط یکی از دو شرط z' و z" آزموده شوند و در صورت برقرار بودن آن گذار متناظر اجرا شود. در خودکارهای تناوبی علاقه‌مندیم علاوه بر انشعاب‌های فصلی انشعاب‌های از نوع عطفی نیز داشته باشیم. در انشعاب‌های عطفی در صورت وجود دو گذار فوق اگر q به صورت عطفی انتخاب کند بایستی هر دو شرط z' و z" را بیازماید و در صورت برقرار بودن هر دو ورودی سیستم باید هر دو مسیر حاوی q' و q" را طی کند. به این ترتیب خودکارهای بوچی و رابین خود می‌توانند از نوع خودکارهای متعارف یا تناوبی تعریف شوند. به عبارت دیگر شرط پذیرش خودکارهای متعارف یا تناوبی از نوع شرط بوچی یا رابین تعریف شود.

دسته دیگری از خودکارها یعنی **خودکارهای بازگشتی مرتبه اول** و بالاتر ارایه شده‌اند که توسیعی از خودکارهای بر درختهای نامتناهی هستند. ایده اولیه این خودکارها از موستوفسکی در [۱۳] و همچنین از امرسون و یوتلا در [۵] است و توسط کیوولا در [۱۲] توسعه یافته است. اثبات شده است که خودکارهای بازگشتی مرتبه اول تناوبی دقیقاً متناظر فرمول‌های حساب Mu هستند. به عبارت دیگر رویه نحوی دقیقی برای ترجمه فرمول‌های حساب Mu به خودکارهای بازگشتی مرتبه اول تناوبی و برعکس وجود دارد [۱۲]. با استفاده از چنین رویه نحوی تصمیم پذیری حساب Mu و اصل پذیری حساب Mu خطی نیز اثبات شده است [۱۲]. به این ترتیب اکنون تبدیل مساله وارسی الگو در حساب Mu به یک مساله در حوزه نظریه خودکارها به ویژه خودکارهای بازگشتی مرتبه اول تناوبی پشتوانه نظری دقیق دارد. در صورت استفاده از چنین تبدیلی مساله وارسی الگو در حساب Mu عبارت است از ارایه الگوریتمی برای وارسی ویژگی‌های توصیف شده در حساب Mu که به خودکارهای بازگشتی مرتبه اول ترجمه شده‌اند در سیستمی که توسط یک ساختار کریپکی مدل شده است. اگر این الگوریتم بر اساس اندازه مدل خطی یا چندجمله‌ای باشد یک الگوریتم مطلوب است.

۶- نظریه خودکارهای بازگشتی مرتبه اول

تعریف ۶-۱ یک خودکار متعارف A بر درختهای n–انشعابی چهارتایی $A=(Q,q_0,\Delta,\Omega)$ است که

- Q مجموعه‌ای متناهی از حالت‌هاست.
- q_0 حالت شروع است.
- رابطهٔ گذار Δ به شکل $\Delta\subseteq Q\times P(Z\cup\overline{Z})\times(Q\setminus\{q_0\})^n$ است و برای هر گذار $(q,Z,\overline{q})$ مجموعه Z متناهی است. ($\overline{q}$ یک رشته n تایی از حالات است).
- Ω شرط پذیرش است. شرط پذیرش بوچی که خودکار متعارف A را به یک خودکار بوچی تبدیل می‌کند شامل مجموعه‌ای از حالت‌های پذیرا است که حداقل یکی از آنها بایستی در یک اجرای خودکار بی نهایت بار تکرار شود. شرط پذیرش رابین شامل مجموعه‌ای از زوجهای پذیرا است که هر زوج خود حاوی مجموعه‌ای از حالت‌های پذیرا و مجموعه‌ای از حالت‌های طرد است. اگر بخواهیم یک اجرای A پذیرا باشد باید بایستی زوج پذیرایی وجود داشته باشد که در آن حداقل یک حالت پذیرا و هیچ حالت طردی بی نهایت بار (در آن اجرا) تکرار شود.

تعریف ۶-۲ یک خودکار تناوبی A بر درختهای n–انشعابی پنج تایی $A=(Q,q_0,\Delta_1,L_1,\Omega)$ است که در آن:

- Q مجموعه متناهی از حالات است.
- q_0 حالت شروع است.
- $\Delta\subseteq Q\times(Q\setminus\{q_0\})$ رابطهٔ گذار است.
- $L:Q\to(\{\wedge,\vee,T\}\cup\{X_i\mid i\in[n]\}\cup Z\cup\overline{Z}$ تابعی است که به هر حالت یکی از علائم $T,\vee,\wedge$، $\perp$ یا X_i یا یک گزارهٔ اتمی یا نقیض یک گزاره اتمی را بر چسب می زند.
- Ω شرط پذیرش است.

هر حالت q از خودکار فوق بایستی محدودیت های زیر را ارضا کند:

۱- اگر $L(q)\in\{T,\perp\}\cup Z\cup\overline{Z}$ آنگاه حالت q' وجود ندارد که $(q,q')\in\Delta$ باشد. در این صورت حالت q را اتمی می نامیم.

۲- اگر برای $i\in[i]$ داشته باشیم $L(q)=X_i$ آنگاه دقیقاً فقط یک و حالت q' وجود دارد به طوری که $(q,q')\in\Delta$ است. در این ورت حالت q را i – حالت می نامیم.

۳- اگر $L(q)\in\{\wedge,\vee\}$ باشد آنگاه حداقل یک حالت q' وجود دارد به طوری که $(q,q')\in\Delta$ است. در این صورت حالت q را $\wedge-$حالت یا $\vee-$حالت می نامیم.

تعریف ۶-۳ فرض کنید $A=(Q,q_0,\Delta,\Omega)$ یک خودکار متعارف بر درختهای –n انشعابی باشد. خودکار A را درخت گونه می نامیم اگر و فقط اگر

- مجموعه حالت‌ها یعنی $Q\subseteq N^*$ تشکیل یک درخت متناهی بدهد.
- حالت شروع q_0 ریشه ε است.
- برای هر گذار $(q,Z,\overline{q})\in\Delta$ و هر i<n حالت $\overline{q}_i$ فرزند یا نیای q است.

اگر $A=(Q,q_0,\Delta_1L_1\Omega)$ یک خودکار تناوبی بر درختهای n– انشعابی باشد آنگاه A را درخت گونه می نامیم اگر و فقط اگر

- $Q\subseteq N^*$ تشکیل یک درخت متناهی بدهد.
- حالت شروع q_0 ریشه ε است.
- برار هر گذار $(q,q')\in\Delta$ حالت q' فرزند یا نیای q است.

در یک خودکار تناوبی یا متعارف درخت گونه حالت q را یک حالت حلقه ای می نامیم اگر و فقط اگر گذار $q'\to q$ وجود داشته باشد که q از نیاکان q' باشد. عمق یک خودکار درخت گونه معادل عمق درخت Q است.

تعریف ۶-۴ یک خودکار بازگشتی مرتبه اول تناوبی یا متعارف بر درختهای –n انشعابی یک خودکار تناوبی یا متعارف A بر درختهای n–انشعابی است به طوری که A خودکاری درخت گونه است و شرط پذیرش Ω برای A به شکل

$\Omega=(G,R)$ است که $G\cap R=f$ و اجتماع G و R مجموعهٔ حالت‌های حلقه ای A است. خودکارهای بازگشتی مرتبه اول را از این به بعد FR- خودکار می‌نامیم.

۷- قضیه اساسی در تناظر فرمول‌های حساب Mu و خودکارهای بازگشتی

تعریف ۷-۱ فرض کنید f یک mKn–فرمول به شکل نرمال مثبت باشد. FR–خودکار تناوبی متناظر فرمول f که آن را با $A(f)$ نشان می دهیم به طریقهٔ زیر ساخته می شود:

درخت متناهی T که توسط زیر فرمول‌های f برچسب گذاری می شود را به طریقه استقرایی زیر تعریف می کنیم:

۱- ریشهٔ T را با خود f برچسب گذاری می کنیم: $T(\varepsilon)=f$

۲- اگر T(t) به یکی از اشکال $y\wedge y'$ یا $y\vee y'$ است آنگاه t دارای دو فرزند است که به ترتیب با y و y' برچسب گذاری می شوند.

۳- اگر T(t) به شکل $X_i y$ است آنگاه t دارای یک فرزند است که با y برچسب گذاری می شود.

۴- اگر T(t) به شکل $sz.y$ است (s یکی از عملگرهای نقطه ثابت است). آنگاه t دارای یک فرزند است که با y برچسب گذاری می شود.

۵- اگر T(t) یک فرمول اتمی است آنگاه t یک برگ خواهد بود.

اکنون با استفاده از درخت T خودکار بازگشتی مرتبه اول تناوبی متناظر فرمول f یعنی $A(f)$ به شکل $A(f)=(Q,q_0,D,L,G,R)$ است که:

۱- برای هر $t\in dom(T)$ داریم: $t\in Q$ مگر آن که $T(t)=z\in Z$ باشد و نیای t' برای t وجود داشته باشد به گونه ای که $T(t')=sz.y$ (برای یک y)

۲- $q_0=e$

۳- $(t,t')\in\Delta$ اگر و فقط اگر t و t' متعلق به Q باشند و t' فرزند t باشد یا t' نیای t باشد و فرزندی از t مانند t'' وجود داشته باشد که t'' یک برگ T است و $T(t'')=z$ و $T(t')=6z.y$ (برای یک y)

۴- برای هر $t\in Q$ داریم

۴-۱- $L(t)=\vee$ اگر و فقط اگر T(t) به یکی از شکلهای $y\vee y'$ یا $sz.y$ باشد.

۴-۲- $L(t)=\wedge$ اگر و فقط اگر T(t) به شکل $y\wedge y'$ باشد.

۴-۳- $L(t)=X_i$ اگر و فقط اگر T(t) به شکل $X_i y$ باشد.

۴-۴- L(t) = z برای اتم z اگر و فقط اگر T(t) = z

۵- $G=\{t\in Q \mid$ T(t) به شکل $\upsilon z.\psi$ است.$\}$

۶- $R=\{t\in Q \mid$ T(t) به شکل $\mu z.\psi$ است.$\}$

قضیه ۱. اگر L(A) مجموعه تمامی رشته‌های پذیرفته شده توسط خودکار A باشد برای هر FR–خودکار تناوبی A بر درختهای n–انشعابی داریم. $L(A)=L(f(A))$. بر عکس برای هر μKn–فرمول ϕ که به شکل نرمال مثبت باشد داریم: $L(f)=L(A(f))$ [۱۲] .

بر اساس قضیه اساسی فوق در حالت کلی رابطه یک به یکی میان FR–خودکارهای تناوبی و μKn–فرمول‌ها وجود دارد و در صورت وجود هر یک دیگری به طریقه ساختنی قابل حصول است. سوالی که در اینجا می‌توان طرح کرد آن است که چه زیرکلاسی از μKn–فرمول‌ها با FR–خودکارهای متعارف (غیر تناوبی) معادل هستند؟ در زیر به این پرسش پاسخ می دهیم.

تعریف ۷-۲ اگر ϕ یک μKn–فرمول محدود شده باشد در این صورت FR–خودکار تناوبی متناظر آن یعنی A(ϕ) که طبق تعریف ۴-۱ حاصل شده است را یک FR–خودکار تناوبی محدود شده می نامیم.

FR–خودکارهای تناوبی محدود شده را می‌توان مستقیما بر اساس اعمال محدودیتهایی بر ساختار خود FR–خودکارهای تناوبی تعریف کرد(نگاه کنید [۱۲] فصل سوم). آنچه برای ما مهم است آن است که هر FR–خودکار تناوبی محدود شده معادل یک FR–خودکار متعارف است. در نتیجه هر FR–خودکار متعارف خود معادل (قابل ساخت) یک μKn–فرمول محدود شده خواهد بود و برعکس هر μKn–فرمول محدود شده معادل (قابل ساخت) یک FR–خودکار متعارف است:

قضیه ۲. اگر A یک FR–خودکار تناوبی محدود شده باشد FR–خودکار متعارف B معادل آن وجود دارد و بر عکس به طوری که L(A)=L(B) [۱۲].

۷-۱ تصمیم پذیری FR–خودکارهای متعارف و تناوبی محدود شده و اشتراک خودکارها

بادر نظر گرفتن تعاریف فوق قضایای زیر به سادگی قابل حصول خواهند بود:

قضیه ۳. اگر A یک FR–خودکار تناوبی محدود شده یا یک FR–خودکار متعارف باشد این پرسش که آیا $L(A)=\emptyset$ ؟ تصمیم پذیر است و این تصمیم گیری در زمان خطی بر اندازه A انجام می گیرد [۱۲].

قضیه۴. فرض کنید $A=(Q_1,q_{01},\Delta_1,(G_1,R_1))$ و $B=(Q_2,q_{02},\Delta_2,(G_2,R_2))$ دو FR–خودکار متعارف باشند FR–خودکار متعارف C به طوری که L(C)=L(A)∩L(B) خودکار $C=(Q,q_0,\Delta,(G,R))$ است که در آن:

۱- مجموعه حالات $Q=Q_1\times Q_2\times\{0,1,2\}$ است.

۲- حالت شروع q_0 همان حالت $(q_{01},q_{02},0)\in Q$ است.

۳- شرط پذیرش $\Omega=(G,R)=(G_1\times G_2\times\{2\},R_1\times R_2\times\{2\})$ است.

۴- رابطه گذار Δ به روش زیر ساخته می شود:

گذار $((r_i,q_j,x),z,(r_m,q_n,y))\in\Delta$ موجود است اگر و فقط اگر

۴-۱. $(r_i,z,r_m)\in\Delta_1$ و $(q_j,z,q_n)\in\Delta_2$ باشند.

۴-۲. عنصر سوم یعنی x و y با استفاده از شرایط پذیرش به شکل زیر معیین می شوند:

- اگر x = 0 و $r_m\in G_1$ است آنگاه y = 1 خواهد بود.
- اگر x = 1 و $q_n\in G_2$ است آنگاه y = 2 خواهد بود.
- اگر x = 2 است آنگاه y = 0 خواهد بود.
- در غیر این صورت y = x خواهد بود [۱] .

قضیه ۵. اگر $A=(Q_1,q_{01},\Delta_1,(G_1,R_1))$ و $B=(Q_2,q_{02},\Delta_2,(G_2,R_2))$ دو FR–خودکار متعارف باشند FR–خودکار متعارف C به طوری که L(C) =L(A) ∩ L(B) است را می‌توان در زمان متناسب با $O(|A|.|B|)$ محاسبه نمود و اندازه خودکار حاصل نیز از همین مرتبه پیچیدگی است.

اثبات: این قضیه نتیجه مستقیم قضایای ۳ و ۴ است. در این قضیه اندازه خودکار متعارف $A=(Q,q_0,\Delta,(G,R))$ مساوی $|A|=|Q|+|\Delta|$ است.

۸- الگوریتمی برای درستی‌یابی ویژگی‌های توصیف شده در حساب Mu با استفاده از نظریه خودکارها

خودکارها می‌توانند به طور مستقیم یا غیر مستقیم در مدل سازی سیستم‌های کامپیوتری به ویژه سیستم‌های واکنشی و همروند به کار روند. در روش اول سیستم واقعی مستقیما توسط یک خودکار مدل می شود که در این مدل مجموعه حالات یا الفبای ورودی (مجموعه گزاره های ممکن) مشخص کننده حالات ممکن سیستم واقعی مدل شده هستند. در روش دوم سیستم واقعی ابتدا توسط یک ساختار کریپکی مدل می شود و سپس این ساختار به یک خودکار متعارف بوچی تبدیل می گردد. این روش غیر مستقیم مبتنی بر قضیه زیر است که به

سادگی از تعریف ساختارهای کریپکی و خودکارهای متعارف بوچی قابل حصول است:

قضیه ۶. هر ساختار کریپکی(M = (S , R , L معادل خودکار متعارف بوچی $A=(Q,q_0,\Delta,F)$ است یعنی L(A) = L(M) که در آن

- مجموعه حالات $Q=S\cup\{q_0\}$ است.

- مجموعه حالات پذیرا F = Q است.

- برای $q,q'\in S$ گذار $(q\ ,Z\ ,q')\in\Delta$ موجود است اگر وفقط اگر $(q,q')\in R$ و $Z=L(q')$ باشند. همچنین برای حالت شروع $S_o\in S$ و $Z=L(S_o)$ گذار $(q_0,Z,S_o)\in\Delta$ موجود است.

با توجه به رویه فوق برای ساخت خودکار متعارف بوچی معادل یک ساختار کریپکی پیداست که قضیه زیر درست است:

قضیه ۷. اگر برای ساختار کریپکی (M = (S , R , L خودکار متعارف بوچی $A=(Q,q_0,\Delta,F)$ معادل آن طبق رویه ساخت فوق ساخته شود آنگاه $|A|=O(|M|)$.

از طرف دیگر خودکارها می‌توانند به طور مستقیم یا غیر مستقیم در توصیف ویژگی‌های سیستم‌های کامپیوتری به ویژه سیستم‌های واکنشی و همروند به کار روند. در روش اول هر ویژگی لازم سیستم مستقیما توسط یک خودکار بوچی متعارف مشخص می گردد. بدیهی است که اگر ویژگی لازم پیچیده باشد ممکن است توصیف مستقیم آن توسط یک خودکار بوچی متعارف بسیار پیچیده تر از توصیف آن ویژگی در زبان یک سیستم منطق زمانی باشد. به ویژه بایستی به این نکته توجه کرد که قدرت بیان برخی سیستم‌های منطق زمانی مانند حساب Mu اکیدا از خودکارهای متعارف بیشتر است. در روش غیر مستقیم ویژگی لازم سیستم ابتدا در زبان یک سیستم منطق زمانی بیان می شود و سپس فرمول حاصل به یک خودکار ترجمه می گردد. این روش را روش مشخص‌سازی منطق‌های زمانی توسط خودکارها می نامیم. قضایای مربوط به این روش تاکنون از دیدگاه منطقی برای منطق زمانی خطی و حساب Mu اثبات شده‌اند.

اکنون می خواهیم الگوریتمی برای مساله وارسی الگو بر اساس روش مشخص‌سازی منطق‌های زمانی توسط خودکارها برای سیستم منطقی حساب Mu ارایه کنیم. در این الگوریتم در واقع با تمهیداتی دو مرحله اول حل مساله وارسی الگو یعنی مرحله مدل سازی سیستم و مرحله توصیف ویژگی لازم آن در یک حوزه و هر دو توسط نظریه خودکارها انجام می گیرد. در حالت کلی مدل سیستم و توصیف ویژگی لازم آن هر دو از نوع خودکارهای بازگشتی مرتبه اول تناوبی که توسیع خودکارهای متعارف هستند می باشند.

اگر A خودکار مدل کننده سیستم و S خودکار توصیف کننده ویژگی لازم آن سیستم باشند در این صورت A ویژگی S را ارضا می‌کند اگر و فقط اگر $L(A)\subseteq L(S)$ و این معادل است با $L(A)\cap\overline{L(S)}=\Phi$. این به معنای آن است که گذاری در A وجود ندارد که در S مجاز نباشد. برعکس اگر $L(A)\cap\overline{L(S)}$ تهی نباشد هر گذار موجود در این مجموعه خود مثال نقضی برای آن ویژگی خواهد بود.

با توجه به آن که طبق قضیه ۴ کلاس خودکارهای بازگشتی مرتبه اول تناوبی محدود شده یا خودکارهای بازگشتی مرتبه اول متعارف تحت عمل اشتراک بسته است و همچنین تهی بودن زبان این خودکارها طبق قضیه ۳ تصمیم پذیر است و با توجه به این که طبق قضیه ۱و تعریف ۴-۲ هر خودکار بازگشتی مرتبه اول تناوبی محدود شده یا خودکار بازگشتی مرتبه اول متعارف معادل یک فرمول محدود شده حساب Mu است اکنون می‌توانیم الگوریتم زیر را برای وارسی الگو بر اساس توصیف ویژگی‌ها در حساب Mu و با استفاده از نظریه خودکارها ارایه کنیم:

الگوریتم وارسی الگو بر اساس توصیف ویژگی‌ها در حساب Mu و با استفاده از نظریه خودکارها:

۱- (مرحله مدل سازی سیستم) سیستم واقعی موجود را مستقیما توسط یک خودکار متناهی مدل کنید. با در نظر گرفتن کلیه حالات به عنوان حالت پذیرا این خودکار متناهی یک خودکار بازگشتی مرتبه اول متعارف بوچی خواهد بود. از طرف دیگر یک سیستم را می‌توان ابتدا توسط یک ساختار کریپکی مدل کرده و سپس با استفاده از روشی که در قضیه ۶ بیان شده است ساختار کریپکی حاصل شده را به یک خودکار بازگشتی مرتبه اول متعارف بوچی تبدیل کرد. در هر صورت خودکار بازگشتی مرتبه اول متعارف بوچی حاصل شده را M می نامیم. دقت کنید که مدل کردن یک سیستم واقعی توسط خودکارهای متناهی و ساختارهای کریپکی روشی متعارف در مدل سازی سیستم‌ها با تعیین سطح خاصی از انتزاع است که هم در متون نظری و هم توسط ابزارهای نرم‌افزاری موجود برای وارسی الگو مانند SPIN مورد استفاده قرار می‌گیرند [۱۷]. (برای مثال‌های عینی از مدل کردن سیستم‌ها توسط خودکارهای متناهی و ساختارهای کریپکی نگاه کنید [۱] فصول ۱، ۲ و ۹).

۲- (مرحله توصیف) ویژگی لازم سیستم را توسط μKn-فرمول ψ بیان کنید. نقیض ψ را به دست آورده و آن را به شکل نرمال مثبت تبدیل کنید. حاصل را ϕ می نامیم یعنی $f=PNF(\neg y)$.

اگر ϕ یک μKn-فرمول محدود شده است (چنان که در تعریف ۳-۶ آمده است) این الگوریتم را ادامه دهید.

۳- (مرحله درستی‌یابی)

۳-۱- فرمول ϕ را طبق تعریف ۷-۱ به خودکار بازگشتی مرتبه اول تناوبی معادل آن یعنی (A(ϕ تبدیل کنید.

۳-۲- طبق قضیه ۴ خودکار بازگشتی مرتبه اول B را بیابید که در واقع اشتراک خودکارهای A و M است یعنی L(B) = L(A) ∩ L(M). دقت کنید که چون خودکارهای متعارف بوچی بر رشته‌های نامتناهی حالت خاصی از خودکارهای بازگشتی مرتبه اول هستند این عملیات اشتراک گیری بین دو خودکار بازگشتی مرتبه اول مجاز است.

۳-۳- با استفاده از قضیه ۳ تهی بودن L(B) را بیازمایید: اگر L(B) تهی است سیستم ویژگی لازم را ارضا می‌کند و اگر L(B) تهی نیست سیستم ویژگی لازم را ارضا نمی‌کند و هر عضو L(B) یک مثال نقض برای آن است.

دقت کنید که اگر قضیه تصمیم پذیری خودکارهای بازگشتی مرتبه اول تناوبی در حالت کلی ثابت شود (نه فقط برای نوع محدود شده آنها) در این صورت الگوریتم فوق برای تمامی μKn-فرمول‌ها (نه فقط برای نوع محدود شده آنها) درست خواهد بود.

اکنون می خواهیم به بررسی پیچیدگی زمان اجرای مرحله درستی‌یابی الگوریتم فوق بپردازیم. فرض کنید سیستمی توسط یک ساختار کریپکی M با اندازه |M| یا یک خودکار متعارف بوچی با اندازه O(|M|) مدل شده و ویژگی لازم آن توسط μKn-فرمول به شکل نرمال مثبت f با عمق k بیان شده باشد. در گام ۳-۱ از الگوریتم فوق μKn-فرمول f را به خودکار بازگشتی مرتبه اول تناوبی معادل آن یعنی ($A(f$ تبدیل می کنیم. پیچیدگی زمان اجرای این گام واندازه خودکار بازگشتی مرتبه اول تناوبی حاصل شده را در قضیه زیر به دست می آوریم:

قضیه ۸. فرض کنید f یک $mKn-$فرمول به شکل نرمال مثبت باشد. FR-خودکار تناوبی متناظر فرمول f که آن را با (A(ϕ نشان می‌دهیم به روشی که در تعریف ۷-۱ آمده است در زمان $O(|f|^{k+1})$ قابل ساخت است و اندازه آن نیز از مرتبه $O(|f|^{k+1})$ است.

اثبات: تعریف ۷-۱ الگوریتمی دو مرحله‌ای برای ساختن FR-خودکار تناوبی معادل هر فرمول f در سیستم منطقی μKn ارایه می‌کند. خودکار مرحله به مرحله با مشخص شدن حالات و گذارهایش ساخته می‌شود. بنابر این زمان ساخت خودکار و اندازه آن با یکدیگر معادل هستند. اگر |M| اندازه مدل، |f| طول رشته f و k

عمق فرمول f بر اساس عملگرهای نقطه ثابت باشد اندازه و زمان ساخت این خودکار را به شکل زیر محاسبه می‌کنیم. در مرحله اول از تعریف ۷-۱ درخت متناهی T که توسط زیر فرمول‌های f برچسب گذاری می‌شود را به طریقه استقرایی ساخته‌ایم. در این درخت هر گره‌ای که برای زیر فرمول‌های به شکل $y \wedge y'$ یا $y \vee y'$ ساخته می‌شود دارای دو فرزند است. به همین ترتیب هر گره‌ای که برای زیر فرمول‌های به شکل $X_i y$ ساخته می‌شود دارای یک فرزند است و هر گره‌ای که برای زیر فرمول‌های به شکل $sz.y$ (s یکی از عملگرهای نقطه ثابت است) ساخته می‌شود حداکثر به تعداد زیرفرمول‌های y که دارای عملگر نقطه ثابت هستند فرزند دلرد. بنابر این در درخت حاصل حداکثر تعداد فرزندان هر گره اندازه کل فرمول است که ما آن را با |f| نشان می‌دهیم. از روش ساخت فوق بدیهی است که ارتفاع این درخت متناسب با عمق عملگرهای نقطه ثابت f یعنی همان فاکتور k است. با توجه به این قضیه که حداکثر تعداد کل گره‌های یک درخت |f| تایی با ارتفاع k از مرتبه $O(|f|^{k+1})$ است بنابراین اندازه و در نتیجه زمان ساخت این درخت همگی از مرتبه $O(|f|^{k+1})$ هستند.

در مرحله دوم از روی درخت فوق FR–خودکار تناوبی متناظر فرمول f را می سازیم. در این مرحله حداکثر به تعداد گره های درخت فوق حالات ماشین خودکار و حداکثر به تعداد یالهای درخت فوق گذارهای ماشین خودکار خواهیم داشت که برای تعیین حضور یا عدم حضور هر یک از آنها جستجویی با اندازه حداکثر تعداد گره‌های درخت لازم است. بنابراین زمان به دست آوردن این خودکار و اندازه آن متناسب با حاصل‌ضرب $O(|f|^{k+1})$ در خودش یعنی $O(|f|^{2k+2})$ است که این فرمول به طور مجانبی همان $O(|f|^{k+1})$ است.

□

در گام ۳-۲ از الگوریتم فوق خودکار بازگشتی مرتبه اول B را می‌یابیم که معادل اشتراک خودکارهای A و M است. از قضیه ۵ می دانیم که زمان ساخت این خودکار $O(|A|.|M|)$ است. با توجه به آن که $|A| = O(|f|^{k+1})$ است بنابراین زمان ساخت این خودکار و اندازه آن از مرتبه $O(|M|.|f|^{k+1})$ خواهد بود.

در گام ۳-۳ از الگوریتم فوق برای خودکار بازگشتی مرتبه اول B تهی بودن زبان آن را می آزماییم. از قضیه ۳ می دانیم که زمان این آزمون متناسب با اندازه B است. بنابراین در نهایت زمان اجرای الگوریتم فوق به شکل زیر حاصل شده است:

قضیه ۹. اگر سیستمی توسط یک ساختار کریپکی M با اندازه |M| یا یک خودکار متعارف بوچی با اندازه $O(|M|)$ مدل شده و ویژگی لازم آن توسط μKn- فرمول محدود شده و به شکل نرمال مثبت f با عمق k بیان شده باشد دراین صورت زمان اجرای مرحله درستی‌یابی الگوریتم فوق از مرتبه پیچیدگی زمانی $O(|M|.|f|^{k+1})$ است.

۹- جمع بندی و نتایج

زبان حساب Mu از دیدگاه قدرت بیان یکی از قوی ترین ابزارهای توصیف را در اختیار متخصصین علوم و مهندسی کامپیوتر قرار داده است. مهمترین ایرادی که به این سیستم می‌توان وارد کرد آن است که فرمول‌ها یا جملات این حساب از دیدگاه شهودی چندان قابل فهم نیستند یعنی با مشاهده آنها نمی‌توان به سادگی ویژگی‌ای که آنها برای سیستم بیان می کنند را فهمید. در مقابل مهمترین وجوه مثبت به کار گیری این حساب آن است که اولا تمامی منطق‌های زمانی متعارف به ویژه PLTL، CTL و CTL^* قابل ترجمه به این حساب هستند و ثانیا روشها یا متدهای مختلفی برای ارائه الگوریتم‌های وارسی الگو در حساب Mu وجود دارند که در سایر منطق‌های زمانی قابل اعمال نیستند.

در حل مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود مجموعه‌ای از الگوریتم‌ها ارائه شده‌اند که هم از دیدگاه نظری و هم از دیدگاه پیاده‌سازی مورد بررسی و تحلیل قرار گرفته اند. این مقاله علاوه بر ارایه نحو و معناشناسی حساب Mu و نظریه عمومی خودکارها، الگوریتمی جدید برای وارسی الگو با استفاده از روش مشخص‌سازی فرمول‌های حساب Mu توسط خودکارها برای زیرنظریه‌ای از آن ارایه نمود.

از دیدگاه نظریه پیچیدگی محاسباتی در تحلیل کارایی هر الگوریتم حل مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود دو عامل ۱) اندازه مدل سیستم و ۲) اندازه و عمق فرمول توصیف کننده ویژگی لازم آن بایستی مورد نظر قرار گیرند. اما چنان که در بخش اول مقاله نشان دادیم از میان این دو عامل اولی از اهمیت بیشتری برخوردار است. اندازه مدل سیستم در عمل اغلب بسیار بزرگ است در حالی که اندازه و عمق فرمول توصیف کننده ویژگی لازم آن سیستم در مقایسه با اندازه مدل بسیار کوچک‌اند. بنابراین علاقه‌مندی اصلی محققین این حوزه الگوریتم‌هایی است که هر چه بیشتر نسبت به اندازه مدل کارا باشند. الگوریتم‌های معمول که مبتنی بر استفاده از روش تکرار یا جستجوی تمامی فضای حالات مساله ارایه شده‌اند دارای زمان اجرای از مرتبه $O((|M||f|)^{k+1})$ هستند که در آنها |M| اندازه مدل، |f| طول رشته f و k عمق فرمول f بر اساس عملگرهای نقطه ثابت است. یعنی زمان اجرای این گونه الگوریتم‌ها از مرتبه چند جمله‌ای نسبت به اندازه مدل و از مرتبه نمایی نسبت به عمق فرمول توصیف‌کننده هستند.

با توجه به پیچیدگی حل مساله وارسی الگو در حساب Mu رویکرد محققین این موضوع بیشتر به زیرکلاسهایی از فرمول‌های این سیستم منطقی است. الگوریتم‌هایی مانند الگوریتم طبیعی که دربخش ۴ این مقاله آن را ارایه کردیم و مساله را در حالت کلی حل می‌کنند یا الگوریتم‌هایی مانند الگوریتم پایه و دولوکس به دلیل نوع محدودیتی که بر فرمول‌ها اعمال می‌کنند همگی دارای پیچیدگی زمانی چند جمله ای (غیر خطی) بر اساس اندازه مدل سیستم و پیچیدگی زمانی نمایی بر اساس عمق فرمول توصیف کننده ویژگی هستند. به کارگیری محدودیتهای مناسب و استفاده از روشهای تلفیقی مانند روش مشخص‌سازی منطق‌های زمانی نقطه ثابت توسط خودکارها این امید را ایجاد کرده است که مساله وارسی الگو وقتی توصیف ویژگی‌ها در حساب Mu انجام شود با کمترین اعمال محدودیت دارای الگوریتمی با زمان خطی بر اساس اندازه مدل است. قبل از این مقالـه دو زیرسیستم حسـاب Mu توسط امـرسون و دیگـران در [۱۰،۹،۸] و زیرسیستم بسیار محدودی در [۱۱] مورد بررسی قرار گرفته‌اند که وارسی الگو برای آنها دارای زمان خطی بر اندازه مدل است. برای الگوریتمی که در این مقاله ارایه شد نیز در قضیه ۹نشان دادیم که درستی‌یابی در زمان خطی بر اندازه مدل قابل انجام است.

اکنون به طور خلاصه به مقایسه قدرت بیان و محدودیت‌های نحوی زیرسیستم‌های در نظر گرفته شده توسط امرسون و دیگران و همچنین زیرسیستم ارایه شده در [۱۱] در مقایسه با زیرسیستم محدود شده‌ای که در این مقاله در نظر گرفته شد می‌پردازیم. در زیرنظریه‌ای که ما درنظر گرفته‌ایم تمامی فرمول‌ها بایستی غیرعطفی و محافظت شده باشند (نگاه کنید بخش ۳ تعریف‌های ۳-۴ تا ۳-۶). در اولین زیرنظریه‌ای که امرسون و دیگران در مراجع [۱۰،۹،۸] در نظر گرفته‌اند در واقع از شرط غیرعطفی بودن فرمول‌ها (تعریف ۳-۴) فقط انتخاب اول یعنی اتمی بودن تمامی فرازهای شرکت کننده در زیرفرمول‌های عطفی در نظر گرفته شده است. با توجه به آن که در زیرنظریه ارایه شده در این مقاله در توسعه این شرط امکان آن که این فرازها اتمی نباشند و در عوض فرمول‌هایی به شکل $X_k y$ باشند را فراهم کرده است بنابراین قطعا قدرت بیان بیشتری از اولین زیرسیستم در نظر گرفته شده امرسون و دیگران دارد. در دومین زیرنظریه‌ای که امرسون و دیگران در[۱۰،۹،۸] در نظر گرفته‌اند تمامی زیرفرمول‌ها یا فرازهایی که در زیرفرمول‌های عطفی وجود دارند به جای اتمی بودن و در توسعه آن بایستی فرمول‌های بسته و بدون متغیر آزاد باشند. این ویژگی موجب این محدودیت می‌شود که عملکرهای نقطه ثابت بر فرازهایی که در زیرفرمول‌های عطفی وجود

دارند تاثیری نداشته باشند. با توجه به آن که در زیرنظریه ارایه شده در این مقاله در توسعه این شرط امکان آن که این فرازها فرمول‌هایی بسته نباشند و در عوض فرمول‌هایی به شکل $X_k y$ باشند را فراهم کرده است بنابراین قطعا قدرت بیان بیشتری از دومین زیرسیستم در نظر گرفته شده امرسون و دیگران دارد. دقت در این نکته لازم است که در تمامی این زیرسیستم‌ها شرط محافظت شده بودن (تعریف ۳-۵) بایستی برقرار باشد. در زیرسیستمی از حساب Mu که در [۱۱] در نظر گرفته شده است فرمول‌ها بایستی فارغ از عملگرهای نقطه ثابت جابجایی باشند یعنی هیچ دو عملکر نقطه ثابتی نبایستی به طور موثر بر یکدیگر به کار روند. این زیرسیستم قبل از زیرسیستم‌های اول و دوم امرسون و دیگران که در بالا ارایه شدند مطرح شده‌اند و نشان داده شده است که حتی از زیرسیستم اول امرسون و دیگران قدرت بیان کمتری دارد [۸].

بنابراین محصول این مقاله الگوریتمی برای وارسی الگو وقتی توصیفات در زیرنظریه‌ای از حساب Mu بیان شوند است که اولا از روش مشخص‌سازی فرمول‌های منطق‌های زمانی نقطه ثابت توسط خودکارها بهره می‌گیرد و ثانیا زمان اجرای آن نسبت به اندازه مدل خطی است و همچنین قدرت بیان زیرنظریه در نظر گرفته شده از سایر زیرنظریه‌هایی که دیگران در نظر گرفته‌اند بیشتر است.

مراجع

[1] E. Clarke, O. Grumberg and D. Peled, *Model Checking*, The MIT Press, 1999.

[2] E. A. Emerson and E. Clarke, "Characterizing Correctness Properties of Parallel Programs as Fixpoints," *Proceedings of 7th Int. Colloquium on Automata, Languages and programming*, LNCS 85, Springer-verlag, 1980.

[3] D. Kozen, "Results on the Propositional Mu-Calculus," *Theoretical Computer Science*, 1983.

[4] E. A. Emerson and C. Lei, "Efficient Model Checking in Fragments of the Mu-Calculus," *IEEE Symposium on Logic in Computer Science*, 1986.

[5] E. A. Emerson and C. Jutla, "Tree Automata, Mu Calculus and Determinacy," *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, 1991.

[6] R. S. Streett and E. A. Emerson, "An Automata Theoretic Decision Procedure for Propositional Mu-Calculus," *Proceedings of Colloquium on Automata, Languages and Programming*, *LNCS 85*, Springer-verlag, 1984.

[7] O. Litchtenstein and A. Pnueli, "Checking That Finite State Concurrent Programs Satisfy Their Linear Specifications," *POPL85*, pp. 97-107, 1985.

[8] E. A. Emerson, C. S. Jutla and A. P. Sistla, "On Model Checking for the μ-calculus and Its Fragments, "*Theoretical Computer Science 258*, pp. 491-522, 2001.

[9] E. A. Emerson, C. S. Jutla and A. P. Sistla, "On Model Checking for Fragments of the Mu-Calculus," *Proceedings of 5th International Conference on Computer Aided Verification (CAV93)*, *LNCS 697*, Springer-Verlag, 1993.

[10] E. A. Emerson, "Model Checking and the Mu-Calculus," *Proceedings of the DIMACS Symposium on Descriptive Complexity and Finite Model*, N. Immerman and P. Kolaitis, eds., American Mathematical Society Press, pp. 185-214, 1997.

[11] R. Cleaveland and B. Steffan, "A Linear Time Model Checking Algorithm for the Alternation Free Modal Mu-Caculus," *Formal Methods in System Design*, vol. 2, no. 2, 1993.

[12] R. Kaivola, *Using Automata to Characterize Fixed Point Temporal Logics*, PhD. Thesis, University of Edinburgh, 1997.

[13] A. Mostowski, "Regular Expressions for Infinite Trees and a Standard Form of Automata," *LNCS 208*, Springer-Verlag, 1985.

[14] D. Long, A. Brown, E. Clarcke, S. Jha and W. Marrero, "An Improved Algorithm for the Evaluation of Fixpoint Expressions," *Proceedings of 6th International Conference on Computer Aided Verification (CAV94)*, *LNCS 818*, Springer-Verlag, 1994.

[15] K. McMillan, *Symbolic Model Checking*, PhD. Thesis, Carnegie-Mellon University, 1992.

[16] R. Bryant, "Graph Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, 1986.

[17] B. Berard, M. Bidoit, A. Finkel and F. Laroussinie, *Systems and Software Verification*, Springer, 2001.

[18] M. Izadi and A. Movaghar, "An Efficient Model Checking Algorithm for a Fragment of μ-Calculus," *The Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE2005)*, 2005.

[۱۹] م. ایزدی، ع. موقر، « رویکردی تازه به درستی‌یابی ویژگی‌های توصیف شده در حساب Mu : استفاده از نظریه خودکارها»، **مجموعه مقالات نهمین کنفرانس بین المللی انجمن کامپیوتر ایران** ، ۱۳۸۱.

[۲۰] م. ایزدی، ع. موقر، «آلگوریتمی جدید برای وارسی الگو در حساب Mu»، **مجموعه مقالات دهمین کنفرانس بین المللی انجمن کامپیوتر ایران** ، ۱۳۸۲.

*این پژوهش تحت قرارداد شماره CS-1383-2-01 توسط پژوهشگاه دانش‌های بنیادی حمایت شده است.

محمد ایزدی مدارک تحصیلی خود را در مقطع کارشناسی مهندسی کامپیوتر در سال ۱۳۷۴ و در مقطع کارشناسی ارشد فلسفه علم در سال ۱۳۷۶ و کارشناسی ارشد مهندسی کامپیوتر (نرم‌افزار) در سال ۱۳۸۱ همگی از دانشگاه صنعتی شریف اخذ نموده است. وی اکنون دانشجوی دکترای مهندسی کامپیوتر در دانشگاه صنعتی شریف و عضو هیات علمی پژوهشگاه علوم انسانی و مطالعات فرهنگی است. زمینه‌های مورد علاقه وی در حوزه علوم کامپیوتر عبارتند از: منطق در علوم کامپیوتر و سمانتیک سیستم‌ها، درستی‌یابی و اعتبارسنجی، سیستم‌های توزیع‌شده و مبتنی بر اجزا، مهندسی نرم‌افزار مولفه‌گرا.

علی موقر رحیم آبادی تحصیلات خود را در مقطع کارشناسی مهندسی برق (الکترونیک) در سال ۱۳۵۶ از دانشکده فنی دانشگاه تهران و در مقاطع کارشناسی ارشد و دکتری مهندسی برق و علوم کامپیوتر به ترتیب در سالهای ۱۳۵۹ و ۱۳۶۴ از دانشگاه میشیگان آمریکا به پایان رساده است و هم اکنون دانشیار دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف می باشد. نامبرده قبل از پیوستن به دانشگاه صنعتی شریف در سالهای ۱۳۶۴ الی ۱۳۶۵ در آزمایشگاه بل ودر سالهای ۱۳۶۶ الی ۱۳۶۸ در دانشگاه میشیگان در آمریکا مشغول به تحقیق و تدریس بوده است. زمینه های تحقیقاتی مورد علاقه ایشان عبارتند از: مدلسازی کارایی و اتکاپذیری، درستی یابی و اعتبار سنجی، شبکه های کامپیوتری و سیستم های توزیع شده بی درنگ.