

حل مسائل برنامه‌ریزی هوش مصنوعی با استفاده از ابزارهای واری مدل

سجاد اسفندیاری^۱، وحید رافع^۲، محمود فرخیان^۳

^۱ دانشجوی دکتری، دانشگاه اراک، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، اراک.

^۲ دانشیار، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه اراک، اراک.

^۳ مربی، دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه شهید چمران، اهواز.

نام نویسنده مسئول:

سجاد اسفندیاری

چکیده

مسائل برنامه‌ریزی هوش مصنوعی، مسائلی هستند که برای رسیدن به هدفشان راه‌های بسیار زیادی پیش‌رو دارند و برای دستیابی به هدف خود که رسیدن سریع‌تر به جواب می‌باشد، غالباً برهوش هیجانی تکیه می‌کنند. برای حل اینگونه مسائل بعضاً می‌توان از واری مدل استفاده کرد. واری مدل فرآیندی است که تمامی حالت‌های ممکن یک سیستم را در قالب یک گراف نمایش می‌دهد و به کمک آن درستی‌یابی را انجام می‌دهد. درستی‌یابی یک محصول بیشترین زمان را در طراحی نرم‌افزار (و سخت افزار) صرف می‌کند. از آنجا که ابزارهای واری مدل در واقع به دنبال یافتن یک حالت خاص در یک فضای حالت بسیار بزرگ و جامع هستند، ما در این پژوهش سعی می‌کنیم چند مسئله معروف برنامه‌ریزی را با ابزار قدرتمند واری مدل، مدل‌سازی و حل کنیم تا کارایی و قابلیت‌های مختلف ابزارها در این مسائل مقایسه شود. به‌منظور ارزیابی ابزارهای نامبرده مسائل را بدون هیچگونه هوشمندی خاصی مدل می‌کنیم و هوشمندی جستجو را بر عهده ابزار می‌گذاریم. درنهایت با آزمایشات مختلف و افزایش سطوح مسائل، کارایی دو ابزار در این حوزه را مورد ارزیابی قرار می‌دهیم و خواهیم دید که ابزار Uppaal یک ابزار مناسب در مسائل برنامه‌ریزی هوش مصنوعی است.

واژگان کلیدی: واری مدل، برنامه‌ریزی هوش مصنوعی، Uppaal، Groove.

مقدمه

با توجه به سرعت پیشرفت در حوزه‌های مختلف علمی و ارتباط تنگاتنگ این پیشرفت با علوم کامپیوتر و ایجاد نرم‌افزارهای پیچیده، لزوم افزایش سرعت و دقت نرم‌افزار بیش از پیش به چشم می‌خورد. صحت و سرعت سیستم‌های کامپیوتری مخصوصاً در انجام عملیات و محاسبات سنگین که بشر قادر به انجام آن نیست یک دغدغه‌ی بزرگ را برای کارشناسان حوزه نرم‌افزار پدید آورده است. اساساً تولید سیستم نرم‌افزاری با خطا همراه است این خطاها در بعضی از سیستم‌ها تأثیری بر زندگی بشر ندارند اما اگر این سیستم‌ها از نوع بحرانی-ایمنی باشد ممکن است با یک خطای کوچک، خسارات مالی و جانی جبران ناپذیری را به بار آورد [۱]. مسائل بحرانی-ایمنی می‌تواند به عنوان یک مسئله برنامه ریزی نمایان شود.

برنامه ریزی خودکار و زمان بندی[□] که با عنوان برنامه ریزی هوش مصنوعی^{□□} (AI Planning) نیز معروف هستند، زیرشاخه‌ای از هوش مصنوعی بحساب می‌آید و برای یافتن جواب می‌بایست از یک حالت اولیه شروع کرده و بعد از عبور از یک یا چند حالت میانی به حالت نهایی برسد حرکت بین این حالات از قاعده خاصی پیروی نمی‌کند و براساس هوش هیجانی حالت‌های بعدی انتخاب می‌شود که در این حالت پیچیدگی زمانی و حافظه‌ای بسیار افزایش خواهد یافت [2]. یکی از راه‌حل‌های مسائل برنامه‌ریزی استفاده از ابزارهای واری مدل است.

واری مدل فرآیندی است که طی آن تمامی حالت‌های امکان‌پذیر یک سیستم مفروض را در قالب یک درخت گسترش می‌دهد. واری کننده با یک روش جستجوی خودکار بطور کامل در فضای بوجود آمده حرکت می‌کند و درستی یا نادرستی ویژگی‌ها را بررسی می‌کند در صورت موفق بودن عملیات واری، یک مثال نقض^{□□□} (شاهد^{□□}) نمایش داده می‌شود. مثال نقض‌ها (شاهد‌ها) بعضی از رفتارهای نامطلوب (یا مطلوب) سیستم را برای اصلاح معایب طراحی در اختیار متخصصان قرار می‌دهد. به منظور انجام واری مدل، ابتدا می‌بایست فضای حالت تولید شده را به ساختار کریپکه[□] تبدیل کنیم. ساختار کریپکه بر روی یک مجموعه متناهی (AP) از گزاره‌های اتمیک^{□□} (تجزیه‌ناپذیر) تعریف می‌شود [3].

برای مورد بررسی قرار دادن یک ویژگی در واری مدل می‌بایست آن را توسط یک روش که منطق زمانی^{□□□} نام دارد آن را توصیف کرد منطق زمانی خطی^{□□□□} (LTL) و منطق درخت محاسباتی^{□□} (CTL) را میتوان به‌عنوان دو نمونه شناخته شده از منطق زمانی معرفی کرد. زمان خطی در LTL بدین معناست که برای هر حالت فقط یک گذر خروجی وجود دارد، در حالی که در CTL ممکن است برای هر حالت، چندین گذر خروجی وجود داشته باشد. به عبارتی دارای ساختار درختی باشد [4]. ما در این پژوهش با توجه به نمایی بودن پیچیدگی منطق زمانی LTL، از ساختار CTL برای بیان ویژگی‌های مسائل برنامه ریزی استفاده می‌کنیم.

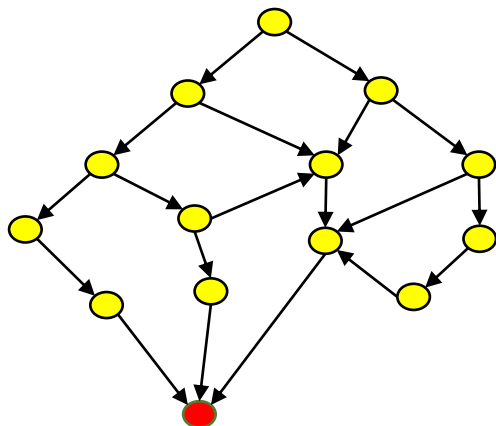
ما در این پژوهش سعی می‌کنیم با ابزارهای Groove و Uppaal مسائل معروف برنامه‌ریزی را مدل‌سازی و حل کنیم. این مسائل عبارت اند از (۱) مسئله پازل هشت‌تایی[□]، (۲) مسئله عبور از پل^{□□}، N-وزیر و نظایر آن. که در این گفتار به دلیل محدودیت صرفاً به شرح دو مسئله پازل هشت‌تایی و مسئله عبور از پل خواهیم پرداخت. ما در این مدل‌سازی تمامی حالت‌های ممکن را برای مسائل در نظر خواهیم گرفت. در واقع مساله را کاملاً غیر هوشمند مدل می‌کنیم و هوشمندی در حرکت را بعهده ابزار خواهیم گذاشت. در نهایت برای ارزیابی دو ابزار واری مدل چند حالت مختلف از پازل هشت‌تایی در نظر می‌گیریم و برای مسئله عبور از پل نیز با افزایش تعداد سربازان این دو ابزار را برای حل مسائل برنامه ریزی مورد ارزیابی قرار می‌دهیم.

ادامه پژوهش به شرح ذیل است: در بخش ۲ مفاهیم مربوط به واری مدل و دو ابزار Groove و Uppaal مورد ارزیابی قرار می‌گیرند. بخش ۳ به مسائل برنامه ریزی و تشریح دو مسئله پازل هشت‌تایی و مسئله عبور از پل اختصاص داده می‌شود. بخش ۴ نحوه شبیه‌سازی دو مسئله پازل هشت‌تایی و عبور از پل در ابزار Uppaal و groove توضیح داده می‌شود. بخش ۵ به ارزیابی دو ابزار groove و Uppaal می‌پردازد و بخش ۶ نیز شامل نتیجه گیری و بحث است.

واری مدل

واری مدل فرآیندی برای ارزیابی یا پذیرش ویژگی‌های یک سیستم است. وقتی یک ویژگی برای دسته‌ای خاص از حالات پذیرفته نشود واری مدل با جستجو در فضای حالت مربوطه یک مثال نقض (یا شاهد) را ارائه می‌کند. در واقع واری مدل از طریق بررسی تمام حالات ممکن مشخص می‌کند که آیا سیستم خصوصیات مربوطه را پشتیبانی می‌کند یا منجر به یک خطا (بعنوان مثال بن‌بست) می‌شود [5]. در واری مدل، بن‌بست به حالتی از فضای حالت گفته می‌شود که هیچ گونه یال خروجی نداشته باشد. عبارت دیگر حالتی که نماینده وضعیتی از سیستم است که پس از آن هیچ سناریو دیگری برای ادامه حیات سیستم وجود نداشته باشد. در شکل (۱) حالت قرمز رنگ دچار بن‌بست شده است.

همانطور که قبلاً اشاره شده برای واریسی ویژگی‌های یک سیستم باید این ویژگی‌های به یکی از دونوع منطق زمانی یعنی منطق زمانی خطی (LTL) یا منطق درخت محاسباتی (CTL) تشریح کرد. در منطق زمانی خطی آینده سیستم به شکل یک گذر حالت در نظر گرفته می‌شود در این منطق برای هر حالت حداکثر یک گذر



شکل (۱): نمایش حالت بن‌بست در واریسی مدل

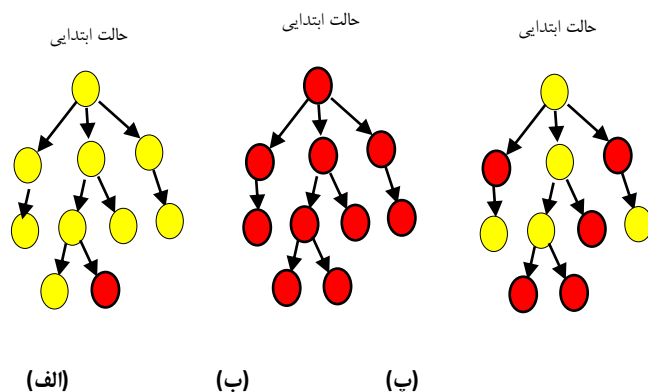
خروجی در نظر گرفته می‌شود. فرموله سازی یک سیستم در منطق زمانی خطی به تعدادی حالت اتمیک نیاز دارد که توسط دو عملگر زمانی X و U و عملگرهای منطقی \neg ، \wedge و \vee با هم ترکیب می‌شوند. در منطق درخت محاسباتی (CTL)، آینده سیستم بشکل یک درخت محاسباتی در نظر گرفته می‌شود به عبارتی، در این منطق برای هر حالت ممکن است بیش از یک گذر خروجی وجود داشته باشد. یک فرمول CTL از مجموعه‌ای از حالت‌های اتمیک، سوره‌های A (برای تمام گذرهای ممکن) سوره‌های E (برای حداقل یک گذر) و عملگرهای زمانی G و F تشکیل شده است. با فرض اینکه q یک گزاره اتمیک است سوره‌های مسیر زیر را خواهیم داشت [4]:

- $A q$: در تمامی مسیرهای شروع شونده از حالت فعلی برقرار باشد.
 - $E q$: در حداقل یک مسیر شروع شونده از حالت فعلی برقرار باشد.
 - عملگرهای زمانی CTL به شرح ذیل هستند:
 - $O q$: q باید در حالت بعدی برقرار باشد.
 - $\langle q \rangle$: q باید حداقل در یک حالت از مسیر برقرار باشد.
 - $[q]$: q باید در تمامی حالات مسیر برقرار باشد.
 - $p U q$: این عملگر در صورتی برقرار است که حداقل در یک حالت از مسیر، q برقرار بوده و در تمام حالت‌های پیش از آن، p نیز برقرار باشد.
 - $!q$: این عملگر در صورتی برقرار است که q برقرار نباشد.
- این فرمول‌ها معمولاً برای واریسی سه ویژگی مهم سیستم‌های نرم‌افزاری استفاده می‌شود:

• **دسترس‌پذیری**: این ویژگی ادعای این را دارد که بالاخره یک حالت خاص مانند q قابل وصول است. مانند بن‌بست. برای نمایش این ویژگی در CTL از $E \langle q \rangle$ استفاده می‌شود (شکل (۲) الف)).

• **ایمنی**: این ویژگی ادعا می‌کند که یک حالت بد هیچگاه اتفاق نمی‌افتد. در CTL برای نمایش ایمنی از $A [q]$ استفاده می‌شود (شکل (۲) ب)).

• **زنده ماندن**: این ویژگی ادعا می‌کند که یک حالت خوب بالاخره باید اتفاق بی‌افتد. زنده ماندن به رو حالت کلی شرطی و غیر شرطی تقسیم می‌شود برای مثال در مورد مسئله «فیلسوف‌های شام‌خور» در حالت شرطی خواهیم گفت "اگر فیلسوفی گرسنه باشد باید حتماً غذا بخورد" و در حالت غیر شرطی می‌گوییم "بالاخره یک فیلسوف باید غذا بخورد". فرمول مسیر $A \langle q \rangle$ با فرض اینکه q یک ویژگی حالت است در CTL برای بیان زنده ماندن قابل استفاده است (شکل (۲) پ)).



شکل (۲): نمایش درختی فرمول‌های مسیر در CTL

ابزار Uppaal

Uppaal یک ابزار برای مدل سازی، شبیه سازی و واریسی سیستم‌های بلادرنگ است که مبتنی بر تکنیک‌های حل مسائل محدود شده (constraint-solving) و on-the-fly است که به طور مشترک توسط دانشگاه اوپسالا^[۵] و دانشگاه آلبرگ^[۶] طراحی و پیاده‌سازی شده است. در واقع این ابزار برای سیستم‌هایی مناسب است که می‌توان آنها را به صورت مجموعه‌ای از فرآیندهای غیرقطعی با ساختار کنترلی و زمان محدود مدل کرد. دو هدف اصلی طراحی Uppaal افزایش کارایی و سهولت استفاده است. در ابزار Uppaal برای سهولت در مدل سازی می‌توان از کد نویسی نیز به همراه حالت گرافیکی برای توصیف مدل استفاده کرد که نحو آن زبان ++C است. Uppaal سیستم‌ها را با استفاده از اتوماتای زماندار مدل می‌کند که در واقع نوعی ماشین متناهی دارای ساعت است. در حالت کلی این ابزار از سه قسمت توصیف سیستم، شبیه‌سازی و واریسی مدل تشکیل شده است که در قسمت توصیف سیستم، رفتار سیستم طراحی می‌شود و قسمت شبیه‌سازی و واریسی مدل برای ارزیابی و واریسی سیستم استفاده می‌شود [6]. این ابزار فرمول‌های مسیر CTL را یا همان نمادهای CTL پشتیبانی می‌کند. یعنی از $E \langle \rangle$ برای دسترس پذیری، از $A []$ برای ایمنی و از $A \langle \rangle$ برای زنده ماندن استفاده می‌شود.

ابزار Groove

پروژه Groove شامل مجموعه ابزارهایی برای مدل سازی ساختار سیستم‌های شی‌گرا در زمان اجرا، کامپایل، طراحی و تبدیل مدل بر پایه تبدیل گراف است که بر استفاده از تکنیک واریسی مدل برای درستی‌یابی سیستم‌های شی‌گرای تمرکز دارد. کاربردهای عمده این ابزار، ارائه فرمالیسم تبدیل مدل و واریسی ویژگی‌ها برای تایید صحت سیستم‌های نرم‌افزاری هستند [7]. در Groove از گراف برای توصیف حالت سیستم و از تبدیلات گراف برای بیان رفتار سیستم استفاده میشود. فضای حالت سیستم با استفاده از یک گراف به‌عنوان حالت اولیه و مجموعه قوانین تبدیل گراف می‌تواند ساخته شود. در ابزار Groove، برای بیان ویژگی‌های حالت، از قوانین تبدیل استفاده می‌شود که دارای سمت چپ و راست یکسانی هستند. برای مثال، فرمول (یا ویژگی) حالت deadlock می‌تواند برای توصیف یک حالت بن‌بست در ابزار Groove به‌کار رود. فرمول‌های مسیر در طول مسیر تعریف می‌شوند. این فرمول‌ها می‌توانند سه ویژگی مهم را در واریسی سیستم‌های نرم‌افزاری به صورت زیر بیان کنند:

- دسترس پذیری $\leftarrow EF q$
- ایمنی $\leftarrow AG q$
- زنده بودن $\leftarrow AF q$

مسائل برنامه‌ریزی در هوش مصنوعی

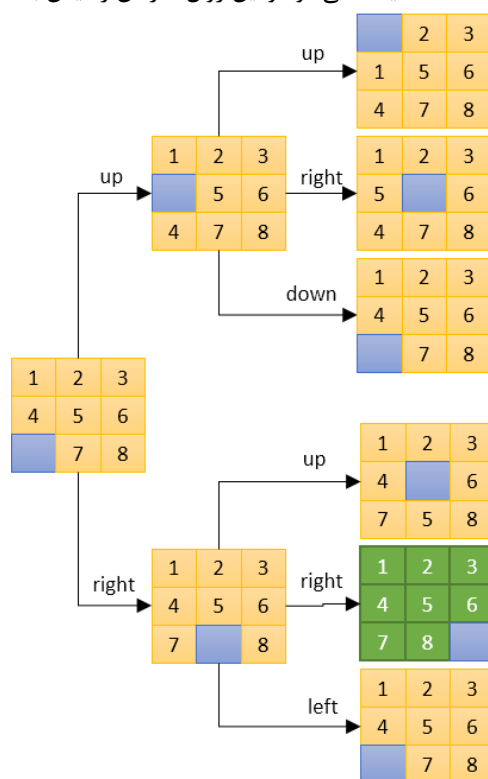
منظور از برنامه‌ریزی در هوش مصنوعی، برنامه‌ریزی دنباله‌ای از فعالیت‌ها برای رسیدن به یک هدف خاص است. عامل‌های حل مسئله مبتنی بر جستجو و عامل منطقی مختلط دو نمونه از عامل‌های برنامه‌ریزی هوش مصنوعی هستند. عامل حل مسئله، می‌تواند دنباله‌هایی از فعالیت‌ها را بیابد که باعث رسیدن به حالت هدف شود، اما این عامل با نمایش تجزیه‌ناپذیر حالت سروکار دارد و در نتیجه برای عملکرد خوب، به ابتکارات خاص دامنه نیاز دارد. عامل مختلط منطقی و گزاره‌ای، بدون ابتکارات خاص

دامنه، میتواند برنامه ریزی‌ها را پیدا کند، زیرا از ابتکارات مستقل از دامنه‌ی مبتنی بر ساختار منطقی مسئله استفاده می‌کند. اما این عامل بر استنتاج گزاره‌ای اولیه (فاقد متغیر) متکی است و این بدان معنی است که وقتی تعداد فعالیت‌ها و حالت‌ها بسیار زیاد باشد، در آنها غرق می‌شود. برای جلوگیری از غرق شدن می‌بایست این عامل بصورت هوشمندانه حرکت کند تا فضای حالت به‌وجود آمده کاهش پیدا کند در ادامه دو مسئله پازل هشت‌تایی و عبور از پل را به‌عنوان مسائل مهم در برنامه‌ریزی تشریح می‌شود.

مسئله پازل هشت‌تایی

در مساله پازل هشت‌تایی ما یک صفحه ۳ در ۳ داریم که با ۸ کاشی با شماره‌های یک تا هشت پر شده‌اند. ما می‌توانیم کاشی‌های مجاور خانه‌ی خالی را هل دهیم و به جای خانه خالی بیاوریم. با این کار ما می‌توانیم جای کاشی‌ها را عوض کنیم. هدف از این بازی این است که کاشی‌ها را با جابه‌جایی به شکل مرتب شده درآوریم (به ترتیب خانه‌های ۱-۸ و درنهایت خانه خالی) و مسیری را که منتهی به جواب می‌شود بدست آوریم.

به عنوان مثال شکل (۳) یک حالت اولیه ساده را نشان می‌دهد همانطور که ملاحظه می‌شود برای هر حالت تمامی مسیرهای ممکن در نظر گرفته می‌شود و برای هر مسیر جدید یک حالت ایجاد می‌شود و این روال تا زمان رسیدن به هدف ادامه دارد.



شکل (۳): حرکت‌های فضای خالی در مسئله پازل هشت‌تایی تا رسیدن به هدف

مسئله عبور از پل

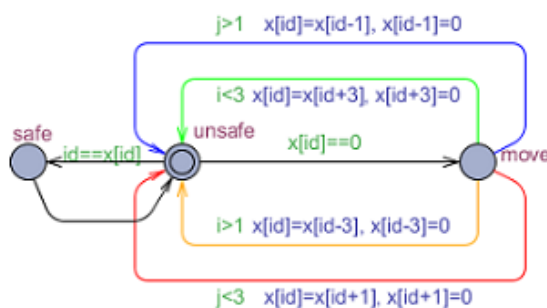
در مسئله عبور از پل تعدادی سرباز قصد دارند شبانه از یکطرف پل (نا امن) به سمت دیگر (امن) حرکت کنند. ظرفیت پل در هر لحظه حداکثر ۲ نفر است. این سربازها یک مشعل دارند که باید در هنگام عبور از پل از آن استفاده کنند. سرعت سربازها برای عبور از پل متفاوت است و هنگامی که دو سرباز قصد عبور از پل را دارند سرباز سریع‌تر می‌بایست سرعت خود را به اندازه سرباز کندتر کاهش دهد این مسئله طوری طراحی شده است که یک سرباز نیز می‌تواند مشعل را در دست گرفته و از پل عبور کند. سرعت این سربازها در جدول (۱) آمده است.

جدول (۱): زمان عبور سربازها از پل

سرباز	۱	۲	۳	۴	۵	۶	۷	۸	۹
زمان	۵	۱۰	۲۰	۲۵	۳۰	۴۵	۶۰	۸۰	۱۰۰

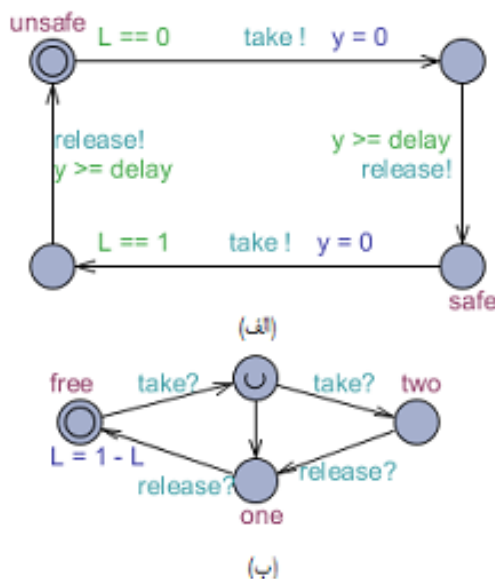
شبه‌سازی

در این بخش ما نحوه مدل‌سازی (شبه‌سازی) مسائل برنامه ریزی را با دو ابزار Uppaal و Groove تشریح می‌کنیم. در ابزار Uppaal برای حل مسئله پازل هشت‌تایی یک ماشین‌تای یک ماشین خواهیم داشت که به تعداد سلول‌ها (نه سلول) آن را تکثیر می‌کنیم و وجه تمایز این ماشین‌ها سطر (i)، ستون (j) و شماره سلول (id) است. سلول خالی را مقدار صفر در نظر می‌گیریم. همچنین برای تعیین مقدار اولیه و تشخیص حالت نهایی از یک آرایه ۹ عنصری (x) استفاده می‌کنیم و زمانی که ترتیب آرایه به ترتیب دلخواه تغییر یافت بازی خاتمه می‌یابد. شکل (۴) نحوه حرکت حاوی مقدار صفر (خالی) را در ابزار Uppaal نشان می‌دهد. ما در این مدل سازی از سه حالت safe (حالتی که سلول در جای خود قرار دارد)، unsafe (حالتی که سلول در جای اصلی خودش قرار ندارد) و move (حالتی است که فقط عنصر با مقدار صفر می‌تواند در آن قرار بگیرد و حرکت بعدی خود را انتخاب کند) استفاده می‌کند حالت move صرفاً برای خوانایی بیشتر مورد استفاده قرار گرفته است. برای سلول حاوی مقدار صفر چهار حرکت (بالا (نارنجی)، پایین (سبز)، چپ (آبی) و راست (قرمز)) وجود دارد. بعد از حرکت سلول صفر در صورتی که مقدار در جای خودش قرار بگیرد می‌تواند به حالت امن برود. برای بررسی حالت هدف می‌بایست از ویژگی دسترس پذیری استفاده کرد در واقع باید به ابزار بگوییم که حالتی را به ما نمایش دهد که در آن تمام سلول‌ها در حالت امن باشند. که این درخواست ویژگی دسترس‌پذیری در CTL است.



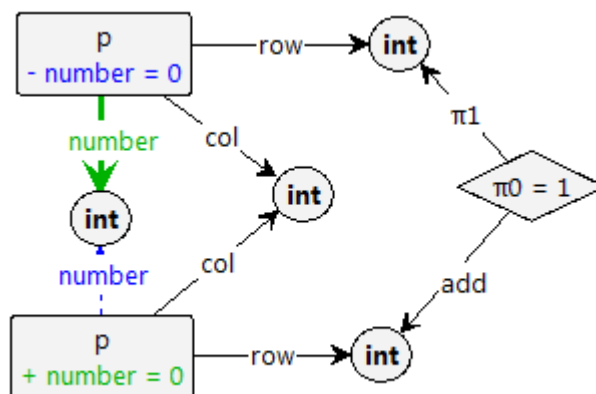
شکل (۴): نحوه حرکت سلول صفر در مسئله پازل هشت‌تایی در ابزار Uppaal

برای حل مسئله عبور از پل با Uppaal نیاز به یک ماشین سرباز (شکل (۵) الف)) و یک ماشین مشعل (شکل (۵) ب)) داریم. که ماشین سرباز به تعداد سربازان کپی می‌شود و وجه تمایز بین آن‌ها سرعتشان (delay) است که از طریق یک کانال با یکدیگر ارتباط برقرار می‌کنند. هر سرباز یک سیگنال (take) به مشعل می‌دهد و در صورت آزاد بودن مشعل، آن را می‌گیرد در هر زمان حداکثر دو سرباز می‌تواند مشعل را بگیرد. هر سرباز برای گذر از حال امن به نا امن باید حتماً مشعل را در اختیار داشته باشند و بعد از عبور از پل باید مشعل را رها کند (release) یکی از ویژگی‌های بارز Uppaal این است که برای عبور از یک گذر به گذر دیگر می‌توان زمان تعیین کرد که در شکل (۵) ب) نحوه بکارگیری آن آورده شده است.



شکل (۵): نحوه طراحی حرکت سرباز و مشعل در Uppaal

برای حل مسئله پازل هشت‌تایی در Groove نیاز است برای هر حرکت به سمت بالا، پایین، چپ یا راست یک قانون نوشته شود شکل (۶) مربوط به حرکت به پایین است که در واقع مقدار سطر حاوی صفر را یک واحد افزایش می‌دهد و محتوای آن را با مقدار سلول پایین‌تر از خود تغییر می‌دهد با این کار سلول حاوی مقدار صفر به سمت پایین حرکت می‌کند برای سه حالت بالا، چپ و راست نیز باید قوانین مربوطه رسم شود ابزار Groove معمولاً برای حل اینگونه مسائل مناسب نیست و دچار انفجار فضای حالت می‌شود. برای غلبه بر این مشکل، [8] با استفاده از الگوریتم شبکه بیزین این ابزار را گسترش داده‌است و ما نیز از این نسخه از ابزار در ارزیابی استفاده می‌کنیم.

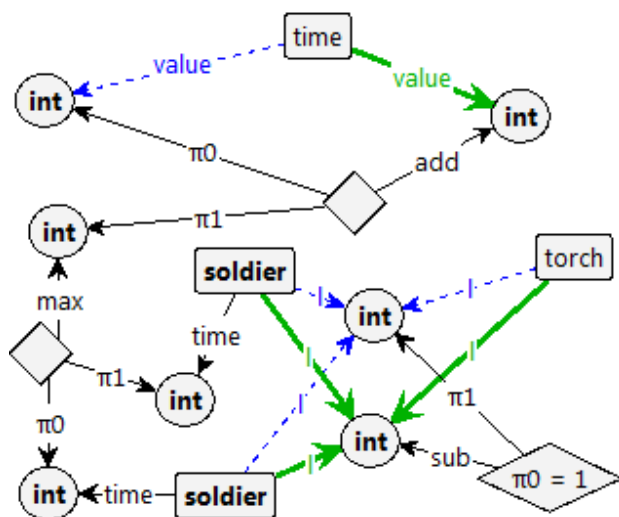


شکل (۶): حرکت به سمت پایین سلول حاوی مقدار صفر در ابزار Groove

آخرین مدل سازی در این پژوهش مربوط به مسئله عبور از پل با ابزار Groove است. مسئله قابل توجه این است که در ابزار Groove برای گذر از حالتی به حالت دیگر، زمان را نمی‌توان لحاظ کرد. برای غلبه بر این مشکل از یک متغییر (زمان) استفاده می‌کنیم و در هنگام عبور از پل مقدار سرباز کندتر را با زمان کل جمع می‌کنیم. شکل (۷) نحوه شبیه‌سازی سرباز را از روی پل در ابزار Groove را نشان می‌دهد. همانطور که ملاحظه می‌شود در این ابزار می‌توان مشعل و سرباز در یک قانون قرار دارند در این شبیه‌سازی دو سرباز که مقدارشان با مشعل یکی است حرکت خود را آغاز می‌کنند و مقدار زمان با مقدار ماکزیمم سرعتشان جمع می‌شود شبیه‌سازی طوری انجام شده است که ممکن است دو سرباز یکی باشند به عبارت دیگر یک سرباز را با خودش انتخاب می‌کند که این عمل شبیه‌سازی حرکت یک سرباز به تنهایی را محقق می‌سازد.

ارزیابی

در این بخش قصد داریم کارایی دو ابزار واریسی مدل Groove و Uppaal را بر روی دو مسئله برنامه‌ریزی هوش مصنوعی (مسئله پازل هشت‌تایی و مسئله عبور از پل) مورد ارزیابی قرار دهیم همانطور که اشاره شده در مدل‌سازی این مسائل هیچ‌گونه هوشمندی برای حرکت و جابجایی عناصر در نظر گرفته نشده است و تمامی حالات ممکن شبیه‌سازی می‌شود در واقع با این کار ما هیچ کمکی به ابزار در راه رسیدن به جواب نکرده‌ایم و خود ابزار با استفاده از تکنیک‌های واریسی مدل به هدف دست می‌یابد. محیط اجرایی ارزیابی‌های انجام شده متشکل از: Windows 7، CPU 2.20GHz core™ i7QM و RAM 6GB است.



شکل (۷): شبیه‌سازی حرکت سرباز برای عبور از پل با Groove

برای ارزیابی مسئله عبور از پل، ۶ حالت را مورد بررسی قرار می‌دهیم حالت اول شامل چهار سرباز با شماره‌های ۱، ۲، ۳ و ۴ است که کمترین مقدار برای عبور این ۴ سرباز ۶۰ واحد زمانی است در ابزار Uppaal می‌توان با انتخاب سریع‌ترین را رسیدن به هدف به این مقدار دست یافت اما برای ابزار Groove بدلیل اینکه این ابزار ویژگی زمان را برای حرکت بین حالات ندارد لذا برای ارزیابی نیاز است که ما مقدار ۶۰ را در زمان واریسی به واریسی کننده اعمال کنیم در واقع باید به واریسی کننده بگوییم که آیا می‌توان در زمان ۶۰ هرچهار سرباز در طرف امن پل قرار بگیرند؟ که در صورت مثبت بودن جواب نحوه حرکت سربازها را به ما نشان می‌دهد. در جدول (۲) مقایسه زمان اجرای مسئله عبور از پل را با ۴، ۵، ۶، ۷، ۸ و ۹ سرباز را نشان داده شده است که برای این نمونه‌ها به ترتیب حداقل زمان‌های ۶۰، ۹۰، ۱۳۰، ۱۷۵، ۲۳۵ و ۳۰۰ وجود دارد که ابزار می‌بایست به دنباله‌ای از مسیرها دست یابد که زمان سپری شده برای عبور تمام سربازان این مقادیر باشد. همانطور که در جدول (۲) ملاحظه می‌شود ابزار Uppaal بسیار سریع‌تر از Groove عمل می‌کند. NA در جدول (۲) نشان‌گر عدم رسیدن به جواب بهینه در ابزار مربوطه است. این ارزیابی نشان می‌دهد که ابزار Groove نمی‌تواند ابزار مناسبی برای حل مسئله عبور از پل باشد. البته باید این موضوع را در نظر گرفت که در ابزار Groove بعد از رسیدن به جواب گراف حالت آن را نیز رسم می‌کند که این امر خود بسیار زمان‌بر است.

جدول (۲): زمان اجرای مسئله عبور از پل

Uppaal (ثانیه)	Groove (ثانیه)	حداقل زمان	ابزار
			سرباز
۰,۰۰	۴,۶۲	۶۰	۴ تا ۱
۰,۰۰	۱۳,۸۲	۹۰	۵ تا ۱
۰,۰۲	۴۰,۰۵	۱۳۰	۶ تا ۱
۰,۰۹	NA	۱۷۵	۷ تا ۱
۰,۲۱	NA	۲۳۵	۸ تا ۱
۰,۹۲	NA	۳۰۰	۹ تا ۱

برای ارزیابی مسئله پازل هشت‌تایی از سه نوع ترکیب ورودی مختلف استفاده می‌کنیم که در جدول (۳) نشان داده شده‌اند. همانطور که ملاحظه می‌شود در این روش نیز ابزار Uppaal بسیار کارآمدتر از ابزار Groove است همچنین نتایج نشان می‌دهد که ابزار Groove در این مسئله نتایج مناسب‌تری را نسبت به مسئله قبل دارد. در این ارزیابی ۵ حالت مختلف از چپ‌دمان اولیه هشت عدد را در نظر گرفته‌ایم. در مورد اول تنها با دو جابجایی پازل تکمیل می‌شود البته باید به این نکته توجه داشت که ابزارهای واریسی مدل کلیه حالات را واریسی می‌کنند تا به هدف برسند. حالت‌های بعدی به مراتب پیچیده‌تر از حالت اول هستند که مقادیر مربوط به زمان رسیدن به هدف در جدول (۳) نشان داده شده است.

جدول (۳): زمان اجرای مسئله پازل هشت تایی

Uppaal (ثانیه)	Groove (ثانیه)	چیدمان اولیه		
		۱	۲	۳
۰,۰۲	۰,۵۲	۱	۲	۳
		۴	۵	۶
		۷	۸	
		۱	۳	
۰,۳	۱,۷۵	۵	۲	۶
		۴	۷	۸
		۲	۳	۶
		۱	۴	۸
۰,۹۴	۷,۷۵	۷	۵	
		۶	۱	۲
		۴	۷	۳
		۵	۸	
۲,۰۲	۵۶,۱۱	۸	۲	
		۴	۷	۳
		۵	۸	
		۸	۲	
۷,۴۱	۱۸۱,۳۲	۴	۷	۳
		۵	۱	۶

نتیجه گیری و کارهای آتی

وارسی مدل یک تکنیک شناخته شده و خودکار برای تایید/نقض ویژگی‌های داده شده در یک سیستم است. از واریسی مدل بعضا در حل مسائل برنامه‌ریزی در هوش مصنوعی استفاده می‌شود مسائل برنامه‌ریزی به مسائلی اطلاق می‌گردد که برای رسیدن به هدف، یک سری مسیر در پیش‌رو دارد. انتخاب مسیر مناسب برای رسیدن به هدف بهینه (کوتاه‌ترین مسیر) پایه و اساس الگوریتمی ندارد و براساس هوش هیجانی صورت می‌پذیرد. ما در این پژوهش سعی کردیم از دو ابزار قوی در واریسی مدل یعنی Groove و Uppaal برای حل دو مسئله برنامه‌ریزی یعنی پازل هشت‌تایی و عبور از پل استفاده کنیم. با مدل کردن این دو مسئله در ابزارهای نامبرده و مقایسه آن‌ها به این نتیجه رسیدیم که ابزار Uppaal بسیار کارآمدتر از ابزار Groove در حل مسائل برنامه‌ریزی است.

نتایج نشان داد که ابزار Uppaal می‌تواند یک ابزار بسیار قدرتمند در حل مسائل برنامه‌ریزی هوش مصنوعی باشد و این امکان وجود دارد که این ابزار حتی از تعدادی از ابزارهایی که صرفا برای برنامه‌ریزی طراحی شده‌اند قوی‌تر باشد لذا مقایسه این ابزار با ابزارهای برنامه‌ریزی هوش مصنوعی می‌تواند به‌عنوان کار آتی این پژوهش پیشنهاد گردد.

منابع و مراجع

- [۱] مرادی، مریم، یوسفیان، رزا، رافع، وحید، "ارائه راهکاری جهت مقابله با مشکل انفجار فضای حالت در سیستمهای تبدیل گراف با استفاده از الگوریتمهای پرندگان و جستجوی گرانشی"، مجله مهندسی برق دانشگاه تبریز، جلد ۴۵، شماره ۴، زمستان ۹۴.
- [2] Ghallab, M, Nau, D. S., Traverso, and P., *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004
- [3] Browne, M. C., Clarke, E. and Grümberg, O., "*Characterizing finite Kripke structures in propositional temporal logic*", *Theoretical Computer Science*, Vol 59, No 1-2, pp. 115-131, 1988.
- [4] Clarke, E. and Emerson, E. A., "*Design and synthesis of synchronization skeletons using branching time temporal logic*", In *Workshop on Logic of Programs*, Springer Berlin Heidelberg, pp. 52-71, 1981.
- [5] Rensink, A., "*The GROOVE Simulator: A Tool for State Space Generation*", pp 479-485, 2003.
- [6] Larsen, K. G., Pettersson, P., Yi, W., "*UPPAAL in a nutshell*", Vol 1, pp. 134-152, 1997.
- [7] Rensink, A., "*The GROOVE Simulator: A Tool for State Space Generation, Applications of Graph Transformations with Industrial Relevance (AGTIVE)*", 3062 *LNCS (2003)*, pp. 479-485, 2003.
- [8] Pira, E., Rafe, V., Nikanjam, A., "*Deadlock Detection in Complex Software Systems Specified through Graph Transformation Using Bayesian Optimization Algorithm*", *Journal of Systems and Software*, Vol 131, pp. 181-200, 2017.

ⁱ Automated planning and scheduling

ⁱⁱ Artificial intelligence planning

ⁱⁱⁱ Counterexample

^{iv} Witness

^v Kripke

^{vi} Atomic propositions

^{vii} Temporal logic

^{viii} Linear temporal logic

^{ix} Computation tree logic

^x 8-puzzle problem

^{xi} The bridge crossing problem

^{xii} reachability

^{xiii} safety

^{xiv} liveness

^{xv} Uppsala

^{xvi} Aalborg