

یک الگوریتم مسیر یاب مازمبتنی بر اتوماتای سلولی دو بعدی

محمد رضا میبیدی شهرام گلزاری

ازمایشگاه سیستم های نرم افزاری

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

تهران-ایران

چکیده: در این مقاله یک الگوریتم مسیر یاب ماز مبتنی بر اتوماتای سلولی (CA) دو بعدی ارائه شده است. هدف از این الگوریتم یافتن مسیری با طول کمینه از سلول مبدأ به سلول مقصد است به طوری که مسیر از موانع عبور ننماید. الگوریتم دارای دو فاز اکتشاف و ردیابی مجدد می باشد. قوانین اتوماتا به صورتی طراحی شده اند که در فاز اکتشاف از سلول مبدأ موجی منتشر شود و در حین انتشار از سلولهایی که می گذرد ردی از خود بر جای گذارد. در فاز ردیابی مجدد از سلول مقصد شروع کرده و رد به جای گذاشته توسط موج را دنبال کرده تا به سلول مبدأ برسیم که مسیر طی شده در این فاز مسیر مطلوب خواهد بود. الگوریتم پیشنهادی ساده بوده و تمام تراکنشهای آن محلی و منطبق با خواص حاکم بر اتوماتای سلولی می باشد. این الگوریتم مسیر مطلوب را در یک CA دو بعدی $m \times m$ در $O(m^2)$ گام زمانی می یابد.

کلمات کلیدی: اتوماتای سلولی، مسیریابی، الگوریتمهای مسیر یاب ماز، طراحی فیزیکی، الگوریتم موازی

۱- مقدمه

الگوریتمهای مسیر یاب ماز کوتاهترین مسیر بین دو نقطه مبدأ^۱ (S) و مقصد^۲ (T) را در یک گراف توری مستطیلی مسطح^۳ پیدا می کنند. به منظور انجام مسیریابی توسط این الگوریتمها ابتدا کل طرح (فضایی که می خواهیم در آن عملیات مسیریابی انجام دهیم) را بسادگی توسط گراف توری مدل می نمایند. برای انجام این کار کل طرح را بوسیله مجموعه ای از سلولهای مربعی با سطح واحد که در یک آرایه دو بعدی قرار گرفته اند، نمایش می دهند. درست مثل اینکه یک توری روی طرح انداخته باشیم.

حال با استفاده از این سلولها و ارتباطات بین آنها گراف را بدست می آورند. بدین صورت، که هر سلول C_i بوسیله یک گره V_i در گراف نمایش داده می شود. گره های متناظر با سلولهایی که شامل مانع هستند را گره های مسدود و بقیه گره ها را گره های غیر مسدود می نامند. در صورتی که سلولهای C_i و C_j با هم همسایه باشند بین گره های V_i و V_j در گراف لبه وجود دارد. وزن هر لبه در گراف برابر با یک در نظر گرفته می شود بجز لبه های مرتبط با گره های مسدود که وزن این گونه گره ها برابر صفر در نظر گرفته می شود. با توجه به توضیحات فوق شکل ۱-ب گراف توری مربوط به طرح شکل ۱-الف را نمایش می دهد.

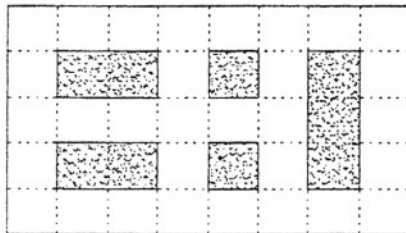
در گراف توری مستطیلی مسطح هر گره با چهار گره همسایه در ارتباط می باشد. گره های متناظر با نواحی از طرح که می توان عمل مسیر یابی را از طریق آنها انجام داد را با حالت غیر مسدود و بقیه گره ها را با حالت مسدود نمایش می دهند. هدف الگوریتمهای مسیر یاب ماز، پیدا کردن مسیری بین گره های مبدأ و مقصد می باشد بطوری که این مسیر از هیچ یک از گره های با حالت مسدود نگذشته و در ضمن کمترین طول را نیز

^۱ Source

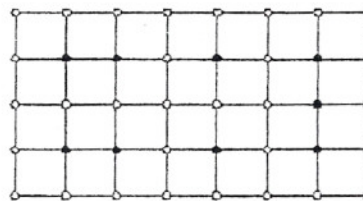
^۲ Target

^۳ Planner Rectangular Grid Graph

دارا باشد. این دسته الگوریتمها دارای دو فاز اکتشاف^۴ و ردیابی مجدد^۵ می باشند و الگوریتم با فاز اکتشاف آغاز می شود. در این فاز از گره مبدأ شروع کرده، تمام مسیرهای ممکن را که از آن آغاز می شوند در نظر گرفته و تمامی آنها را ادامه می دهیم تا یکی از آن مسیرها به گره مقصد برسد.



الف



ب

شکل ۱: گراف توری

هنگامی که گره مقصد مورد دسترسی قرار گرفت، فاز ردیابی مجدد آغاز می شود. در این فاز با استفاده از روش بازگشت به عقب^۶ مشخص می شود که کدام گره های گراف در مسیر متصل کننده گره های مبدأ و مقصد قرار دارند. پس از انجام این فاز الگوریتم خاتمه می یابد. برای پیاده سازی فاز ردیابی مجدد لازم است که در فاز اکتشاف اطلاعاتی در مورد مسیرها در هر یک از گره ها قرار گیرد و سپس با استفاده از این اطلاعات فاز ردیابی مجدد انجام می گیرد [7,8,11,15]. الگوریتمهای مسیریاب ماز در یافتن مسیر ربات و فاز مسیریابی از طراحی فیزیکی VLSI کاربرد دارند [8,15].

اولین الگوریتم از این دسته الگوریتمها توسط Lee معرفی شده که کوتاهترین مسیر را در یک گراف توری با ابعاد $h \times w$ در زمان $O(h \times w)$ می یابد [8] و پس از آن الگوریتمهای متعددی بر مبنای الگوریتم Lee و برای بهبود در نحوه گسترش مسیر و زمان اجرای الگوریتم گزارش شده است که از آنجمله می توان به الگوریتمهای Soukup [14] و Hadlock [7] اشاره نمود. یکی از خصوصیات جالب الگوریتمهای مسیریاب ماز، مستتر بودن نوعی توازی در آنها می باشد. این نکته موجب شده است که این الگوریتمها بر راحتی بر روی ساختارهای موازی نگاشت شده و در عین ثابت بودن پیچیدگی الگوریتم، زمان اجرای الگوریتم به دلیل انجام شدن الگوریتم توسط سخت افزار مناسب و کم شدن تعداد دستورالعملهای انجام شده توسط هر پردازنده تا حد زیادی کاهش یابد. تعدادی از این نگاشتها توسط Sagar و Massara گزارش شده است [19].

در این مقاله یک الگوریتم مسیریاب ماز مبتنی بر CA ارائه می شود. الگوریتم مطرح شده بسیار ساده می باشد. در این الگوریتم هر سلول CA دستورالعملهای بسیار اندکی را اجرا می کند که موجب کم شدن زمان اجرای الگوریتم می شود. ساختار مورد استفاده نیز دارای خواصی نظیر سادگی، توازی و محلی بودن می باشد که متناسب با الگوریتمهای مسیریاب ماز است. در این الگوریتم ابتدا طرح را بر روی CA دوبعدی نگاشت نموده و سپس با استفاده از قوانین ساده و محلی مناسب فازهای مسیریابی را پیاده سازی می نماییم. در این مقاله ابتدا در بخش ۲ در مورد مفاهیم اساسی CA صحبت خواهیم نمود و سپس در بخش ۳ به شرح الگوریتم و نحوه پیاده سازی آن توسط CA می پردازیم.

۲- اتوماتای سلولی (CA)

یک شبکه منظم از ماشینهای با حالت محدود را در نظر بگیرید. به هر یک از این ماشینها، سلول^۸ می گوئیم. هر کدام از این سلولها طبق الگوی ثابت و یکسانی با بعضی از سلولهای مجاور خویش در ارتباط هستند. این ارتباط محلی بوده و برای تمام سلولها یکسان است. خود سلول و تمام سلولهایی که با آن در ارتباط هستند، مجموعه همسایگان^۹ سلول را تشکیل می دهند. هر سلول در هر لحظه از زمان می تواند یکی از حالتها^۷ موجود در مجموعه حالت را اختیار نماید. مجموعه حالت برای تمامی سلولها یکسان می باشد. در هر لحظه از زمان حالت تمامی سلولها به صورت همزمان و بر اساس یک قانون تغییر می کند. این قانون خود تابعی از حالتها^۸ همسایگان سلول بوده و بنابراین در هر لحظه از زمان، حالت بعدی هر سلول به حالت فعلی تمامی همسایگانش بستگی دارد. این شبکه از یک پیکره بندی اولیه شروع به کار کرده، در هر مرحله زمانی با اعمال قانون به تمامی سلولها پیکره بندی بهنگام شده و با گذشت زمان شبکه یک رفتار پیچیده و جالب از خود تولید می نماید.

^۴ Exploration

^۵ Retrace

^۶ Back Tracking

^۷ Run Time

^۸ Cell

^۹ Neighborhood

با توجه به توضیحات فوق، وجه تمایز CA نسبت به دیگر شبکه های اتوماتی عبارت است از: ساده بودن ساختار، محلی بودن ارتباطات

بین سلولها، برقراری الگوی ارتباطی یکسان برای تمام سلولها، بهنگام سازی همزمان سلولها، بهنگام سازی سلولها توسط قانون یکسان و تولید رفتارهای جالب و پیچیده از سلولهای ساده. با توجه به این خصوصیات در اکثر مستندات از CA به عنوان ساختاری میزازی، محلی و یکسان^{۱۱} نام برده شده است که پراختی می تواند رفتارهایی را که دارای این سه ویژگی هستند، شبیه سازی نماید[4,6,9,15,18]. امروزه CA در زمینه های متعددی از جمله تولید الگوهای تصادفی، نظریه محاسبات، مدل سازی سیستم های فیزیکی و بیولوژیکی و محاسبات کاربردی مورد استفاده قرار گرفته است[3,5,6,9,12,15,16,18].

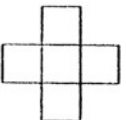
یک نمونه از CA هایی که امروزه متداول می باشد، CA دوبعدی است. در این ساختار سلولها در یک شبکه دو بعدی قرار دارند. یکی از

نگات مهم در این CA نوع همسایگی می باشد. دو نوع همسایگی مهم در این ساختار عبارتند از همسایگی von Neumann و همسایگی Moore.

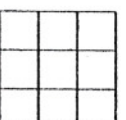
در همسایگی Moore برای هر سلول هشت سلول همسایه و در همسایگی von Neumann چهار سلول همسایه در نظر گرفته می شود(شکل ۲). در هر دو همسایگی سلول قرار گرفته در مرکز همسایگی بهنگام می شود. Packard تمامی خصوصیات CA دوبعدی را بررسی نموده است [10].

تا کنون با استفاده از روشهای مختلفی اثبات شده است که CA یک مدل محاسباتی عمومی می باشد[1,2] ولی در عمل از این مدل اغلب

برای شبیه سازی پدیده های فیزیکی پویا استفاده شده و این مدل بیش از آنکه مورد توجه دانشمندان علم کامپیوتر قرار گیرد، توجه پژوهشگران سایر علوم مانند فیزیک و بیولوژی را به خود مشغول نموده است. آنها تکامل تدریجی پدیده های فیزیکی را با استفاده از قوانین CA بصورت گام به گام شبیه سازی می نمایند. در صورتی که طبق تعریف یک مدل محاسباتی مکانیزمی برای توصیف تکامل تدریجی پدیده ها نمی باشد بلکه ساختاری است که برای انجام محاسبات بروی داده های ورودی بکار می رود.



همسایگی Moore همسایگی von Neumann



شکل ۲: دو نوع همسایگی در CA دوبعدی

در این مقاله به CA از دید یک ماشین محاسباتی عمومی نگریسته شده و الگوریتمی برای حل یک مسئله توسط CA ارائه شده است. برای حل مسایل محاسباتی به یک ساختمان داده و یک رویه نیاز می باشد. ساختمان داده ورودی و خروجی را در خود نگه داشته و از رویه برای تبدیل ورودی به خروجی استفاده می شود. مراحل پردازش و تبدیل ورودی به خروجی در CA توسط انتقال حالتیهای CA پیاده سازی می شود. هر چند در تعریف CA استاندارد، حافظه منظور نشده است ولی اگر CA خواسته باشد نقش یک ماشین محاسبه گر عمومی را بازی کند اولاً هر سلول اتوماتا نیاز به تعدادی حافظه برای نگهداری مقادیر ورودی و خروجی داشته و ثانیاً اتوماتا باید قابلیت خواندن مقادیر ورودی و قابلیت نوشتن مقادیر خروجی (از به) حافظه را داشته باشد. تعریف ارائه شده در زیر که در این مقاله بعنوان تعریف CA در نظر گرفته شده است، اصلاح شده تعریف اتوماتای میلی^{۱۲} می باشد.

CA یک $(Q, d, V, \Sigma, \Delta, \delta, \lambda)$ می باشد که در آن:

۱- مجموعه حالاتی است که هر سلول می تواند اختیار کند.

۲- d ابعاد فضای سلولی را مشخص می نماید. اگر $d = 2$ باشد در این صورت یک CA دوبعدی خواهیم داشت.

۳- برای هر سلول x در CA، $\{v_0, x + v_1, L, x + v_k\}$ مجموعه مشخص کننده $k+1$ همسایه ای می باشد که

بصورت مستقیم با سلول در ارتباطند.

۴- Σ فضای ورودی CA می باشد.

۵- Δ فضای خروجی CA می باشد.

۶- δ تابع انتقال است که بفرم $Q \rightarrow Q \times \Sigma^n$ می باشد. بر اساس تابع انتقال حالت بعدی هر سلول به حالت و مقادیر حافظه

ای داده می تمامی همسایگان آن سلول در مرحله فعلی بستگی دارد. n تعداد رجیسترهای ورودی و خروجی هر سلول است.

۷- رابطه میل^{۱۳} است که زیر مجموعه متناهی از $\Delta^n \times \Sigma^{k+1}$ می باشد. این میل مقدار هر حافظه خروجی سلول را با

توجه به حالت و مقادیر حافظه ای ورودی همسایگان مشخص می سازد. در این جا هر سلول CA در همان حافظه هایی می نویسند که از آنها می خواند و در نتیجه $\Delta = \Sigma$ می باشد.

^{۱۰} Uniform

^{۱۱} Mealy Automaton

^{۱۲} Transducer Relation

۳- یک الگوریتم مسیریاب ماز برای CA دو بعدی

۳-۱- ایده الگوریتم

اساس الگوریتم پیشنهادی مانند دیگر الگوریتمهای مسیریاب ماز، الگوریتم Lee می باشد. بر اساس این الگوریتم در فاز اکتشاف از نقطه مبدأ موجی منتشر می شود. این موج در حین انتشار از محلهایی که می گذرد، ردی از خود برجای می گذارد. در محلهایی که موج با مانع (سلولهای مسدود) برخورد می کند، حرکت موج از آن نقاط متوقف شده در حالی که حرکت موج از نقاط دیگر ادامه می یابد. زمانی که موج به مقصد می رسد فاز اکتشاف به پایان رسیده است، ولی موج همچنان به حرکت خود ادامه می دهد.

در فاز ردیابی مجدد این بار از مقصد شروع کرده و رد به جای گذاشته توسط موج را دنبال می کنیم تا به مبدأ برسیم. مسیر طی شده در فاز ردیابی مجدد همان مسیر مطلوب خواهد بود که کوتاهترین مسیر از مبدأ تا مقصد می باشد. با توجه به آنکه هر دو فاز اکتشاف و ردیابی مجدد با تراکنشهای محلی انجام پذیر است، لذا با توجه به خصوصیت اصلی CA (محلی بودن تراکنشها) پیاده سازی الگوریتم توسط CA براحتی امکان پذیر می باشد.

۳-۲- پیاده سازی الگوریتم توسط CA

در این قسمت نحوه پیاده سازی الگوریتم توسط CA تشریح می شود. به این منظور می بایست ابتدا کل طرح در یک CA دوبعدی و با همسایگی Von Neumann نگاشت شود. در این روش کل طرح به صورت مجموعه ای از سلولها با مساحت واحد در نظر گرفته می شود. سلولهایی که موانع طرح را می پوشانند با حالت اولیه Block(B) در نظر می گیریم. حالت سلولهای متناظر با نقاط مبدأ و مقصد را به ترتیب برابر Source(S) و Target(T) قرار داده و حالت بقیه سلولها را Free(F) قرار می دهیم. بدیهی است که مسیر مورد نظر از میان سلولهای با حالت F می گذرد. بنابراین برای هر سلول به یک متغیر حافظه ای بنام State که نشاندهنده حالت آن سلول می باشد نیاز داریم. از این به بعد منظور از سلول B، سلول S، سلول T و سلول F به ترتیب سلول با حالت Block، سلول با حالت Source، سلول با حالت Target و سلول با حالت Free خواهد بود.

همانطور که توضیح داده شد در فاز اکتشاف می بایست از سلول مبدأ موجی تولید شود. برای پیاده سازی تولید موج می توان بصورت زیر عمل نمود. در گام اول موج یاد شده تنها به سلولهای همسایه با سلول مبدأ می رسد. حال سلولهای F که موج به آنها رسیده است حالت خود را به Mark(M) تغییر می دهند. در گام بعد موج تمام سلولهای F که دارای یک همسایه M هستند را در بر می گیرد که این سلولها نیز حالت خود را M می کنند. عمل انتشار موج به همین منوال ادامه یافته و با این روش فاز اکتشاف پیاده سازی می شود. بنا به سناریوی فوق می توان انتشار موج را توسط قانون ساده زیر پیاده سازی نمود: هر سلول F که دارای حداقل یک همسایه M یا S باشد، حالت خود را به M تغییر می دهد.

همانطور که در ایده الگوریتم ذکر شد، موج می بایست در محلهایی که از آن عبور می کند، ردی از خود بجای گذارد تا در فاز ردیابی مجدد، مسیر با توجه به آن ایجاد شود. در صورتی که سناریوی فوق این عمل را انجام نداده و تنها موج را منتشر می کند. حال باید طرحی اندیشید که همگام با انتشار موج، رد انتشار موج نیز در سلولها ذخیره گردد.

همانطور که دیدیم در هر مرحله که موج گسترش پیدا می کند تعدادی از سلولهای F تبدیل به M می شوند، در حقیقت سلولهای F که دارای همسایه های M یا S هستند، تبدیل به M می شوند. لذا برای ذخیره کردن رد موج کافی است بدانیم کدام همسایه سلول F، حالت M یا S داشته و این اطلاع را در یک متغیر حافظه ای نگهداری کنیم. به این منظور از یک متغیر حافظه ای بنام Direction استفاده می کنیم. این متغیر می تواند یکی از مقادیر Up, Down, Left, Right و Null را اختیار نماید. در وضعیت اولیه مقدار این متغیر برای همه سلولهای F و T برابر Null و برای بقیه سلولها بی اهمیت است.

حال هر سلول F که دارای همسایه M یا S باشد، چک می کند آن همسایه در کدام طرف سلول قرار دارد و با توجه به آن متغیر Direction خود را مقدار می دهد. بدین صورت که اگر همسایه M یا S در بالا، پایین، چپ یا راست سلول باشد متغیر Direction به ترتیب مقادیر Left, Up, Down, و Right را اختیار می نماید. در صورتی که سلول دارای بیش از یک همسایه M باشد متغیر Direction آن به دلخواه مقدار متناظر با یکی از همسایه های M را اختیار می کند. سلول T نیز عملی مشابه با سلول F انجام می دهد، با این تفاوت که حالت آن تغییر نمی کند و فقط متغیر Direction آن مقدار می گیرد.

زمانی که موج به سلول T می رسد (متغیر Direction این سلول مقداری مخالف Null می گیرد)، فاز اکتشاف به پایان رسیده و فاز ردیابی مجدد شروع می شود. در این فاز می بایست مسیر بصورت گام به گام از سلول T تا سلول S مشخص شده و سلولهای واقع شده در مسیر حالت Path(P) بخود بگیرند. حال ببینیم این مسیر چگونه ایجاد می گردد. برای توضیح این مطلب ابتدا مفهوم زیر را تعریف می کنیم:

• تعریف ۱: نشانه رفتن سلول A به سلول B: گوییم سلول A به سلول B نشانه رفته است اگر یکی از موارد زیر اتفاق افتاده باشد:

- سلول A همسایه پایینی سلول B باشد و مقدار Direction سلول A برابر Up باشد.
- سلول A همسایه بالایی سلول B باشد و مقدار Direction سلول A برابر Down باشد.
- سلول A همسایه سمت چپ سلول B باشد و مقدار Direction سلول A برابر Right باشد.
- سلول A همسایه سمت راست سلول B باشد و مقدار Direction سلول A برابر Left باشد.

در فاز ردیابی مجدد اگر سلول T یا P دارای یک همسایه M باشد و در ضمن به طرف آن همسایه M نیز نشانه رفته باشد، آنگاه آن همسایه M حالت P به خود می گیرد. نتیجه اعمال مطلب فوق به این صورت خواهد بود: زمانی که متغیر Direction مربوط به سلول T مقدار مخالف Null گرفت (یعنی زمانی که موج به سلول T رسید) فاز ردیابی مجدد آغاز گشته است. بنابراین در اولین گام از فاز جدید، آن سلول M که سلول T به طرف آن نشانه رفته است حالت P به خود می گیرد و در گام بعد نیز آن سلول M که سلول P جدید به آن نشانه رفته است تبدیل به P می شود و این عمل تا رسیدن به سلول S ادامه می یابد. سلولهای P مسیر مورد نظر را تشکیل می دهند. زمانی الگوریتم عملاً انجام یافته تلقی می شود که مسیر مورد نظر به سلول S برسد. (یعنی سلول S دارای یک همسایه T یا P باشد)

۳-۳- ساختار هر سلول

در الگوریتم پیشنهادی هر سلول دارای دو متغیر حافظه ای بنامهای State و Direction می باشد. ♦ متغیر State: حالت هر سلول را مشخص می کند و می تواند یکی از مقادیر زیر را اختیار نماید:

- Source(S): سلول مبدأ دارای حالت S می باشد.
- Target(T): سلول مقصد دارای حالت T می باشد.
- Block(B): سلولهایی که مسیر نمی تواند از آنها گذر کند دارای حالت B می باشند.
- Free(F): سلولهایی که مسیر می تواند از آنها بگذرد، دارای حالت F می باشند.
- Mark(M): سلولهایی که موج به آنها می رسد دارای حالت M می شوند.
- Path(P): سلولهایی که در مسیر نهایی قرار می گیرند دارای حالت P می شوند.

♦ متغیر Direction: این متغیر رد انتشار موج در هر سلول را مشخص می کند و می تواند یکی از مقادیر زیر را انتخاب نماید.

- Up: سلولهایی که موج از طریق همسایه بالایی به آنها رسیده است، مقدار متغیر Direction آنها برابر با Up می شود.
- Down: سلولهایی که موج از طریق همسایه پایینی به آنها رسیده است، مقدار متغیر Direction آنها برابر با Down می شود.
- Left: سلولهایی که موج از طریق همسایه سمت چپ به آنها رسیده است، مقدار متغیر Direction آنها برابر با Left می شود.
- Right: سلولهایی که موج از طریق همسایه سمت راست به آنها رسیده است، مقدار متغیر Direction آنها برابر با Right می شود.
- Null: سلولهایی که موج به آنها نرسیده است، مقدار متغیر آنها برابر با Null است.

۳-۴- وضعیت اولیه CA

در وضعیت اولیه می بایست طرح بر روی CA نگاشت شود. به این منظور سلولهای که در آنها پلاک قرار دارد، دارای حالت B می شوند. سلولهای متناظر با مبدأ و مقصد بترتیب حالت S و T بخود می گیرند و حالت بقیه سلولها F خواهد بود. مقدار متغیر Direction در وضعیت اولیه برای سلولهای F و T برابر Null و برای بقیه سلولها بی اهمیت خواهد بود.

۳-۵- وضعیت نهایی CA

در پایان الگوریتم سلولهایی که بر روی مسیر قرار دارند حالت P خواهند داشت. سلولهایی که در وضعیت اولیه دارای حالتهای B, T یا S باشند تا وضعیت نهایی بدون تغییر باقی می مانند.

۳-۶- قوانین CA

- اگر سلولی دارای حالت S باشد، در گام بعد سلول بدون تغییر باقی می ماند.
- اگر سلولی دارای حالت B باشد، در گام بعد سلول بدون تغییر باقی می ماند.
- اگر سلول دارای حالت F باشد، وضعیت های زیر برای آن متصور است:
 - هیچ یک از همسایگان سلول M یا S نباشند، در این صورت سلول بدون تغییر باقی می ماند.
 - فقط یکی از همسایگان سلول حالت M یا S داشته باشد، در این صورت حالت سلول M شده و سلول به سمت همسایه مورد نظر نشانه می رود.
- توضیح: اگر همسایه با حالت S یا M در یکی از جهات بالا، پایین، چپ یا راست سلول قرار داشته باشد، متغیر Direction سلول بترتیب مقادیر Up, Down, Left, Right را اختیار خواهد نمود.
- بیش از یک همسایه سلول دارای حالت M یا S باشد. در این صورت حالت سلول M شده و سلول به دلخواه یکی از همسایگان دارای حالت M یا S را انتخاب نموده و به آن نشانه می رود.
- اگر سلول دارای حالت T باشد، از قوانین سلول با حالت F پیروی کرده با این تفاوت که حالت سلول بدون تغییر باقی می ماند.
- اگر سلول دارای حالت M باشد، وضعیت های زیر برای آن متصور است:
 - هیچ کدام از همسایگان سلول دارای حالت P یا T نباشند، در این صورت حالت سلول بدون تغییر باقی می ماند.

- یکی از همسایگان سلول دارای حالت P یا T باشد، در این صورت سلول تست می کند که آیا همسایه مذکور به سمت سلول نشانه رفته است یا نه. در صورت مثبت بودن جواب حالت سلول P شده و در غیر این صورت سلول حالت سلول بدون تغییر باقی می ماند.
- اگر سلول دارای حالت P باشد، در گام بعد سلول بدون تغییر باقی می ماند.

۷-۳ پیچیدگی الگوریتم

مسیر از مبدأ تا مقصد در یک CA دو بعدی $m \times m$ به هیچ طریقی نمی تواند طولی بیش از m^2 داشته باشد، زیرا مسیر از هر سلول بیش از یکبار نمی تواند عبور کند و تعداد سلولها نیز m^2 می باشد و لذا پیچیدگی الگوریتم از مرتبه $O(m^2)$ می باشد.

۸-۳ اثبات الگوریتم

می خواهیم اثبات کنیم مسیری که توسط الگوریتم فوق ایجاد می شود کوتاهترین مسیر بین مبدأ و مقصد خواهد بود. برای این منظور ابتدا قضایای زیر را اثبات می نماییم. در این قضایا منظور از فاصله یک سلول تا سلول مبدأ، طول کوتاهترین مسیر از آن سلول تا سلول مبدأ می باشد.

قضیه ۱: $P(t)$: اولاً تمام سلولهایی که در وضعیت اولیه دارای حالت F هستند و فاصله آنها تا سلول مبدأ t می باشد، تا گام $t-1$ دارای حالت F بوده و در گام t حالت M را به خود می گیرند.

ثانیاً: تمام سلولهایی که در گام $t-1$ دارای حالت F بوده و در گام t حالت M به خود می گیرند، نسبت به سلول مبدأ دارای فاصله t می باشند.

برهان: قضیه فوق را توسط استقرا تام ثابت می کنیم:

پایه استقرا: $P(1)$ صادق است. زیرا اولاً روشن است که سلولهایی که در وضعیت اولیه ($t=0$) حالت F دارند و فاصله آنها تا سلول مبدأ برابر یک است همگی همسایه سلول مبدأ هستند و طبق قوانین CA در گام یک حالت M بخود می گیرند. و ثانیاً سلولهایی که در گام صفر (وضعیت اولیه) دارای حالت F هستند و در گام یک حالت M بخود می گیرند، طبق قوانین CA لزوماً همسایه سلول مبدأ می باشند (در غیر اینصورت حالت M بخود نمی گرفتند) و فاصله همسایگان سلول مبدأ تا سلول مبدأ نیز برابر یک می باشد.

فرض استقرا: $P(1) \wedge P(2) \wedge P(3) \wedge \dots \wedge P(k)$

حکم استقرا: $P(k+1)$

ابتدا قسمت اول حکم استقرا را ثابت می کنیم: سلولهایی که در وضعیت اولیه حالت F دارند و فاصله آنها تا سلول مبدأ $k+1$ است، بطور قطع تا گام k حالت F دارند زیرا در غیر این صورت عددی مانند r ($0 < r < k-1$) خواهیم داشت که آن سلول تا گام $k-r-1$ ، F بوده و در گام $k-r$ ، M شده است. در این صورت طبق قسمت ثانیاً گزاره $P(k-r)$ ، فاصله آن سلول تا سلول مبدأ $k-r$ می باشد. در حالی که فاصله سلول تا سلول مبدأ برابر $k+1$ است. پس ثابت می شود که سلول تا گام k حالت F دارد.

حال باید ثابت کنیم که در گام $k+1$ حالت سلول M خواهد شد. روشن است که سلولی که فاصله اش نسبت به سلول مبدأ $k+1$ می باشد (مانند سلول A) حداقل یکی از همسایگانش دارای فاصله k نسبت به سلول مبدأ می باشد. طبق قسمت اولاً $P(k)$ این همسایه در گام k حالت M به خود گرفته و بنابراین چون سلول A در مرحله $k+1$ دارای یک همسایه با حالت M می باشد طبق قوانین CA حالت M به خود می گیرد. پایان اثبات قسمت اولاً $P(k+1)$.

حال می بایست قسمت ثانیاً از حکم استقرا ثابت شود:

فرض می کنیم سلول A در گام k ، دارای حالت F بوده و در گام $k+1$ حالت M به خود گرفته است. این وضعیت تنها زمانی اتفاق می افتد که در گام k یکی از همسایگان سلول A (مانند سلول B) دارای حالت M باشد، زیرا در غیر این صورت طبق قوانین CA حالت سلول A برابر M نمی شود.

می توان نشان داد که فاصله سلول A تا سلول مبدأ عددی بزرگتر از k است. زیرا در غیر این صورت عددی مانند r ($0 < r < k-1$) وجود دارد که فاصله سلول A تا سلول مبدأ برابر با $k-r$ است. در آن صورت طبق قسمت اولاً گزاره $P(k-r)$ ، در گام $k-r-1$ حالت سلول A ، M می شود. حالت سلول A تا گام k نیز M باقی می ماند. در حالی که طبق فرض در گام k دارای حالت F می باشد.

همچنین می توان نشان داد که فاصله سلول B تا سلول مبدأ عددی کوچکتر یا مساوی k می باشد. زیرا فرض کرده بودیم که در گام k سلول B دارای حالت M است. در این صورت عددی مانند r ($0 < r < k-1$) وجود دارد که سلول B تا گام $k-r-1$ دارای حالت F بوده و در گام $k-r$ حالت M به خود گرفته است. بنابراین طبق قسمت ثانیاً گزاره $P(k-r)$ ، فاصله آن سلول تا سلول مبدأ عددی کوچکتر یا مساوی k است.

از طرف دیگر سلول A و سلول B همسایه می باشند و روشن است که اختلاف فاصله دو سلول همسایه تا سلول مبدأ برابر یک می باشد. از آنجا که فاصله سلول A تا سلول مبدأ عددی بزرگتر از k و فاصله سلول B تا سلول مبدأ عددی کوچکتر یا مساوی k است، تنها در صورتی سلول A و B می توانند با هم همسایه باشند که فاصله سلول B تا سلول مبدأ برابر k و فاصله سلول A تا سلول مبدأ برابر $k+1$ باشد. که این همان اثبات قسمت ثانیاً حکم استقرا می باشد.

تعریف ۲: رسیدن موج به یک سلول در زمان t : گوییم موج در زمان t به سلول A رسیده است اگر یکی از وضعیتهای زیر اتفاق بیفتد:

۱- سلول A در زمان $t-1$ دارای حالت F باشد و در زمان t حالت M بخود گیرد.

۲- سلول A در زمان t-1 دارای حالت T بوده و مقدار متغیر Direction آن برابر Null باشد و در زمان t متغیر Direction آن مقداری غیر از Null داشته باشد.

قضیه ۲: اگر موج در گام t به سلول T برسد، مسیر ایجاد شده توسط فاز ردیابی مجدد دارای طول t خواهد بود.
 برهان: طبق قضیه ۱ تمام سلولهای T یا F که در فاصله d از سلول مبدأ قرار دارند، در گام d موج از طریق یکی از همسایگانشان به آنها می رسد و بنابراین سلول به سوی همسایه منتقل کننده موج نشانه می رود. روشن است همسایه ای که سلول به آن نشانه رفته در زمان d-1، M، شده و بنابراین فاصله اش تا سلول مبدأ برابر d-1 است.
 بر اساس استدلال فوق اگر موج در گام t به سلول T برسد، آنگاه سلول T به یک سلول M که فاصله اش تا سلول مبدأ برابر t-1 است اشاره می کند و این سلول M در اولین گام از فاز ردیابی مجدد، حالت P به خود می گیرد. این سلول نیز به نوبه خود به سلولی که فاصله اش تا سلول مبدأ t-2 است اشاره می کند و در گام بعد این سلول نیز P می شود و این عمل تا رسیدن به سلول S ادامه می یابد که t گام طول خواهد کشید و در هر گام یک سلول M، p خواهد شد و بنابراین مسیر ایجاد شده دارای طول t خواهد بود. حال می خواهیم اثبات نماییم که مسیر ایجاد شده توسط الگوریتم کوتاهترین مسیر است. با استفاده از قضایای فوق این مطلب به سادگی قابل اثبات است. اگر کوتاهترین مسیر از سلول مبدأ تا سلول مقصد دارای طول d باشد طبق قضیه ۱ موج در زمان d به سلول مقصد می رسد و طبق قضیه ۲ اگر موج در زمان d به سلول مقصد برسد، مسیر ایجاد شده توسط فاز ردیابی مجدد دارای طول d است که همان کوتاهترین مسیر است.

۳-۹- ردیابی الگوریتم

در شکل ۳، چگونگی تغییر حالت سلولها در هنگام اجرای الگوریتم نمایش داده شده است. در این شکل از حروف P، M، T، S و B ترتیب برای نمایش سلولهای با حالت Source، Target، Mark و Path استفاده شده است و اگر در سلولی از هیچ حرفی استفاده نشده باشد بدین معنی است که حالت سلول برابر Free می باشد. در این شکلها برای نمایش مقادیر Up، Down، Left، Right و متغیر Direction سلولها بترتیب از نمادهای \rightarrow ، \leftarrow ، \uparrow ، \downarrow استفاده شده است. در صورتی که مقدار متغیر Direction سلولی برابر Null باشد، هیچ کدام از این نمادها را نخواهد داشت.

۴- نتیجه گیری

در این مقاله یک الگوریتم مسیریاب ماز مبتنی بر اتوماتای سلولی (CA) دو بعدی ارائه شد. این الگوریتم مسیری با طول کمینه از سلول مبدأ به سلول مقصد را می یابد به طوری که مسیر از موانع عبور ننماید. الگوریتم پیشنهادی ساده بوده و تمام تراکنشهای آن محلی و منطبق با خواص حاکم بر اتوماتای سلولی می باشد. هر سلول اتوماتا نیز ساختار ساده ای داشته و در هر لحظه از زمان تنها به محتویات سلولهای همسایه دسترسی دارد. این الگوریتم مسیر مطلوب را در یک CA دو بعدی $m \times m$ در $O(m^2)$ گام زمانی می یابد.

۵- مراجع

- [1] W. Burks, Essays on Cellular Automata, Urbana, IL: University of Illinois Press, 1970.
- [2] J. H. Conway, E. Berlekamp, and R. Guy, Winning Ways for Your Mathematical Plays, Vol. 2, Academic Press, 1982.
- [3] K. Culik, L. Hurd, and S. Yu, "Computation Theoretic Aspects of Cellular Automata," Physica D, Vol. 45, pp. 357-378, 1990.
- [4] D. Farmer, T. Toffoli, and S. Wolfram, Cellular Automata Proceedings of an Interdisciplinary Workshop, Amsterdam, North Holland, 1984.
- [5] L. Gordillo, and V. Luna, "Parallel Sort on a Linear Array of Cellular Automata," IEEE Transaction on Computers, pp. 1904-1910, 1994.
- [6] A. H. Gutowitz, Cellular Automata, Cambridge, MA: MIT Press, 1990.
- [7] F. O. Hadlock, "A Shortest Path Algorithm for Grid Graph," Networks, 1977.
- [8] C. Y. Lee, "An Algorithm for Path Connections and its Applications," IRE Transactions on Electronic Computers, 1961.
- [9] M. Mitchel, "Computation in Cellular Automata: A Selected Review," Technical Report, Santa Fe Institute, Santa Fe, New Mexico, 1996.
- [10] N. Packard, "Two-Dimensional Cellular Automata," Journal of Statistical Physics, Vol. 30, pp. 901-942, 1985.
- [11] Y. Pan, Y.C. Hsu, and W. J. Kubitz, "A Path Selection Global Router," Proceedings of Design Automation Conference, 1987.
- [12] P. Sarkar, "Brief History of Cellular Automata," ACM Computing Surveys, Vol. 32, No. 1, 2000.
- [13] N. A. Sherwani, Algorithm for VLSI Physical Design Automation, Western Michigan University Kluwer Academic Publishers, 1993.



مجموعه مقالات

هفتمین

کنفرانس سالانه

انجمن کامپیوتر ایران

۷ تا ۹ اسفند ماه ۱۳۸۰

مرکز تحقیقات مخابرات ایران



انجمن کامپیوتر ایران
Computer Society of Iran

