اتوماتای یادگیر سلولی

محمدرضا میبدی حمید بیگی مسعود طاهرخانی

مرکز تحقیقات انفورماتیک، آزمایشگاه پردازش موازی

دانشکدهٔ مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

تهران ایران

meybodi@ce.aku.ac.ir beigy@ce.aku.ac.ir

چکیده

در این مقاله مدل جدیدی تحت عنوان اتوماتای یادگیر سلولی معرفی شده و رفتار آن از طریق شبیه سازی کامپیوتری مورد مطالعه و ارزیابی قرار می گیرد. بر اساس این مدل، مجموعه ای از اتوماتون های یادگیر در شبکه ای نظیر آنچه در اتوماتای سلولی وجود دارد، قرار گرفته اند و نسبت به رفتار همسایه های خود عکس العمل نشان می دهند. هر اتوماتان در این مدل، بر اساس یک الگوریتم یادگیری عمل نموده و در طی زمان یاد می گیرد که چگونه بهترین عمل را در مقابل پاسخ محیط پیرامون خود انجام دهد. این مدل پیشنهادی دارای کاربردهای متعددی از جمله مدل کردن سیستم های اجتماعی، سیستمها ی بیولوژیکی، و پردازش تصاویر می باشدکه در گزارش های آینده ارایه خواهد گردید.

کلمات کلیدی: اتوماتای سلولی، اتوماتای یادگیر، اتوماتای یادگیر سلولی

۱- مقدمه

اتوماتای سلولی در اواخر دهه ۱۹۴۰ توسط John von Neumann مطرح و سپس توسط ریاضیدانی بنام Stanislaw Ulam به عنوان مدلی برای بررسی رفتار سیستم های پیچیده پیشنهاد شد[1][2]. اتوماتای سلولی سیستم های دینامیکی هستند که ظرفیت اطلاعاتی پائینی داشته و بهمین دلیل برای برخی کاربردها مطلوب نمی باشند. ظرفیت اطلاعاتی آنها را می توان بکمک یادگیری افزایش داد[3] [4].

اتوماتای یادگیر[1] اولین بار توسط M. L. Tsetlin در اوایل دهه ۱۹۶۰ در کشور شوروی سابق مطرح شد. اتوماتون های یادگیر در محیطی تصادفی عمل نموده و قادر هستند که بر اساس ورودیهای دریافت شده از محیط، احتمال انجام عملیات خود را بروز در آورده تا بتوانند از این طریق کارآیی خود را بهبود بخشند.

در این مقاله، با توجه به ویژگیهای دو مدل فوق، مدل جدیدی بنام اتوماتای یادگیر سلولی[2] (CLA) پیشنهاد می گردد. در این مدل، هر سلول در اتوماتای سلولی مجهز به یک اتوماتان یادگیر می باشد که وضعیت این سلول را مشخص می کند. هر اتوماتان یادگیر، بر اساس یک الگوریتم یادگیری عمل نموده و در طی زمان یاد می گیرد که چگونه از طریق تعامل (فعل و انفعال) با محیط بهترین عمل را انتخاب نماید. در حالت خاص، مدل جدید رفتار اتوماتای سلولی را از خود نشان می دهد و بهمین دلیل CLAدر برگیرنده CA مباشد[5] .

در ادامه، ابتدا به شرح اتوماتای سلولی و اتوماتای یادگیر می پردازیم. سپس مدل جدید اتوماتای یادگیر سلولی را معرفی می کنیم. در پایان، برخی از قوانین استفاده شده را بررسی نموده و نتایج بدست آمده از انجام آزمایشات با این قوانین و اتوماتون های مختلف را ارائه می نماییم.

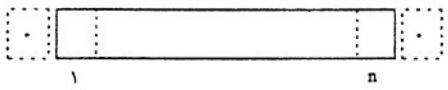
۲- اتوماتای سلولی

اتوماتای سلولی مدلی ریاضی برای سیستم هایی است که در آنها چندین مولفه ساده برای تولید الگوهای پیچیده با هم همکاری میکنند. اتوماتای سلولی از یک شبکه منظم سلولها تشکیل شده است که هر سلول می تواند k ($k > 1$) مقدار مختلف به خود بگیرد. سلولهای اتوماتای سلولی در زمانهای گسسته بطور همزمان و بر طبق یک قانون محلی بنام Φ بهنگام میشوند که در آن مقدار هر سلول بر اساس مقادیر سلولهای همسایه تعیین میگردد. اتوماتای سلولی براساس معیارهای مورد بررسی به دسته های مختلف تقسیم میگردد. بعنوان نمونه بر اساس معیار بعد شبکه، اتوماتای سلولی به اتوماتای سلولی یک بعدی، دو بعدی و غیره تقسیم می گردد و براساس مقدار k به اتوماتای سلولی دودوئی(k=2) و اتوماتای سلولی چند مقداره (k>2) تقسیم میشوند. همچنین اتوماتای سلولی را براساس شکل همسایه ها

[1] Learning Automata (LA)

[2] Cellular Learning Automata (CLA)

میتوان به دو دسته اتوماتای سلولی *با مرز پریودیک* و اتوماتای سلولی *با مرز غیر پریودیک* تقسیم نمود. بعنوان مثال در اتوماتای سلولی یک بعدی با n سلول که در شکل زیر نشان داده شده است در صورتیکه سلولهای شماره ۱ و n را باهم همسایه در نظر گیریم اتوماتای سلولی را با مرز پریودیک و در صورتیکه برای سلولهای ۱ و n دو سلول مجازی با مقدار صفر بعنوان همسایه در نظر گیریم آنرا اتوماتای سلولی با مرزغیر پریودیک مینامند.



شکل ۱: اتوماتای سلولی یک بعدی

در اتوماتای سلولی یک بعدی مقدار سلول i (برای $1 \le i \le n$) در زمان t که با نشان $a_i(t)$ داده میشود بصورت زیر محاسبه میگردد.

$$a_i(t+1) = \Phi[a_{i-1}(t), a_i(t), a_{i+1}(t)]$$

در رابطه فوق، اگر قانون Φ فقط به مقدار همسایه ها بستگی داشته باشد آنرا قانون general مینامند و اگر قانون Φ تابعی از مجموع مقادیرسلولهای همسایه ها و سلول مرکزی باشد آنرا قانون totalistic میگویند و بصورت زیربیان میشود.

$$a_i(t+1) = \Phi[a_{i-1}(t) + a_i(t) + a_{i+1}(t)]$$

درصورتیکه قانون تابعی از مجموع مقادیرسلولهای همسایه و مقدار سلول مرکزی باشد آنرا قانون Outer totalistic میگویند و بصورت زیر نشان داده میشود.

$$a_i(t+1) = \Phi[a_i(t), a_{i-1}(t) + a_{i+1}(t)]$$

همچنین قانون Φ میتوان بصورت قطعی یا احتمالی باشد. در صورتیکه Φ یک تابع تصادفی باشد آنرا قانون احتمالی و در غیر اینصورت آنرا قانون قطعی میگویند.

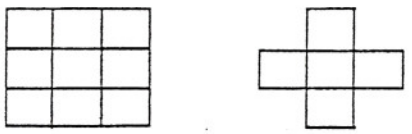
قانون Φ را میتوان به شکلهای مختلفی نشان داد. یک شیوه معمول برای نمایش قانونهای general، استفاده از یک شماره برای قانون است. در این شیوه، شماره قانون برابر است با معادل دهدهی مقدار جدید سلول مرکزی به ازای تمام ترکیبات مختلف سلولهای همسایه. بعنوان مثال در اتوماتای سلولی یک بعدی شماره قانون ۹۰ بصورت زیر میباشد

$(a_{i-1}(t).a_i(t).a_{i+1}(t))$	(۱،۱،۱)	(۱،۱،۰)	(۱،۰،۱)	(۱،۰،۰)	(۰،۱،۱)	(۰،۱،۰)	(۰،۰،۱)	(۰،۰،۰)
$a_i(t+1)$	۰	۱	۰	۱	۱	۰	۱	۰

که عدد ۹۰ معادل دهدهی عدد (۰۱۰۱۱۰۱۰) میباشد. روش دیگری که برای نمایش این دسته از قوانین بکار میرود عبارت منطقی به شکل مجموع حاصلضرب استاندارد[۳] است.

برای نمایش قوانین totalistic از شیوه دیگری استفاده میگردد که در آنها سلولها برای تعیین مقدار یک سلول رای گیری میکنند. در این دسته از قوانین، سلول مرکزی عضوی از مجموعه همسایگان خود در نظر گرفته می شود. در این شیوه از نمایش، قانون را بصورت N-M نمایش میدهند که $M=\{M_1,M_2,...\}$ و $N=\{N_1,N_2,...\}$ میباشد. دراین شیوه از نمایش، قانون بصورت زیر تفسیر میگردد. *اگر مقدار سلول مرکزی* **یک** *باشد و تعداد همسایه هایی که مقدار آنها* **یک** *است متعلق به مجموعه* N *باشد مقدار سلول مرکزی برابر* **صفر** *خواهد شد و اگر مقدار سلول مرکزی* **صفر** *باشد و تعداد همسایه هایی که مقدار آنها* **یک** *است متعلق به مجموعه* M *باشد مقدار سلول مرکزی برابر* **یک** *خواهد شد.* بعنوان مثال قانون ۵۶۷۸۹-۰۱۲۳۴ که قانون اکثریت نامیده می شود بصورت زیر بیان می شود. *اگر مقدار سلول مرکزی* **یک** *باشد و تعداد همسایه هایی که مقدار آنها یک است بزرگتر یا مساوی ۵ باشد مقدار سلول مرکزی برابر* **صفر** *خواهد شد و اگر مقدار سلول مرکزی* **صفر** *باشد و تعداد همسایه هایی که مقدار آنها یک است کوچکتر از ۵ باشد مقدار سلول مرکزی برابر* **یک** *خواهد شد.*

مطالب مطرح شده فوق را میتوان به اتوماتای سلولی با ابعاد بیشتر نیز تعمیم داد. بعنوان مثال برای یک اتوماتای سلولی دوبعدی میتوان چندین ساختار برای شبکه و همسایه ها در نظر گرفت که در این مقاله تنها شبکه های مربعی و دو ساختار زیر برای همسایه ها در نظر گرفته شده است.



شکل ۲: انواع همسایگی های مهم در اتوماتای سلولی

قوانین اتوماتای سلولی دودوئی را بر اساس رفتارشان در طی یک دوره طولانی به چهار کلاس مجزا میتوان تقسیم کرد [9 -6].

کلاس I: کلاس I رفتار *limit point* یا بسیار کسل کننده دارند و بسوی یک وضعیت همگون با شروع از هر حالت اولیه می روند. بطوز مثال مقدار همه سلولها صفر و یا مقداز همه سلولها همه یک میشوند. این کلاس شامل قوانین ۰، ۴، ۱۶، ۳۲، ۳۶، ۴۸، ۵۴، ۶۰ و ۶۲ میباشد

کلاس II: کلاس II رفتار *limit cycle* یا کسل کننده دارند و نهایتا مثل یک فیلتر، ساختارهای ساده، جدا و پریودیک می سازند. ساختارهای ساده ایجاد شده یا پایدار هستند یا پریودیک که در اینصورت، معمولا دوره پریود کوتاهی دارند. گاهی اوقات نیز الگوهایی بوجود می آورند که به سمت راست یا چپ شیفت داده می شوند. این کلاس شامل قوانین ۸، ۲۴، ۴۰، ۵۶ و ۵۸ می باشد.

کلاس III: کلاس III به حالتهای غیرپریودیک و غیرقابل پیش بینی از نظر فضا و زمان[۴] منجر می گردند و رفتار جالبی دارند. اکثر قوانین در این کلاس بوده که شامل قوانین ۲، ۶، ۱۰، ۱۲، ۱۴، ۱۸، ۲۲، ۲۶، ۲۸، ۳۰، ۳۴، ۳۸، ۴۲، ۴۴، ۴۶ و ۵۰ هستند.

---

[۳] Disjunctive Normal Form

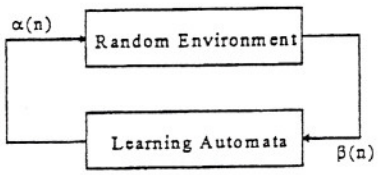
[۴] Chaotic

کلاس IV: در نهایت، مهمترین کلاس مربوط به آن قوانینی است که متعلق به هیچ یک از سه کلاس فوق نبوده و رفتار پیچیده ای از خود نشان میدهند. قوانین این کلاس رفتار بسیار جالب دارند و ساختارهای منتشر شونده و گاهی با عمر طولانی ایجاد می کنند. ادعا شده است که اینگونه قوانین دارای ویژگی جهان محاسباتی می باشند. این کلاس شامل قوانین ۲۰، ۵۲ می باشد.

اتوماتای سلولی دارای کاربردهای فراوانی است. برخی از این کاربردها عبارتند از: تولید کننده اعداد تصادفی[10]، رمزنگاری [11-13]، انتشار شایعه[14]، پردازش تصویر[16,15]، مطالعه زبانهای رسمی [8]، و مرتب سازی اعداد[17].

۳- اتوماتای یادگیر

۳-۱- اتوماتای یادگیر[5]: اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی می گردد و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل خود برای مرحله بعد انتخاب می کند[19,18]. شکل ۳ ارتباط بین اتوماتون یادگیرو محیط را نشان می دهد.



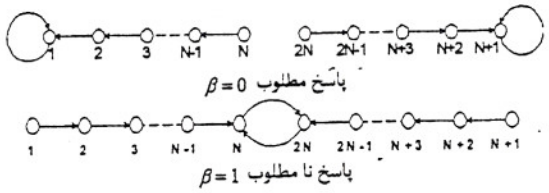
شکل ۳: ارتباط بین اتوماتا یادگیر و محیط

۳-۲- محیط[6]: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که درآن $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \cdots, c_r\}$ مجموعه احتمالهای جریمه میباشد. هرگاه β مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته میشود. در محیط از نوع Q، $\beta(n)$ می تواند بطور گسسته یکی از مقادیر محدود در فاصله [0,1] را اختیار کند و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [0,1] است.

c_i احتمال آنکه عمل α_i نتیجه نامطلوب[7] داشته باشد می باشد. در محیط ایستا[8] مقادیر c_i بدون تغییر باقی می مانند، حال آنکه در محیط غیر ایستا[9] این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیر به دو گروه تقسیم می گردند:

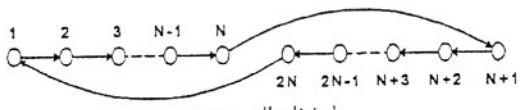
۳-۳- اتوماتای یادگیر با ساختار ثابت[10]: اتوماتای یادگیر با ساختار ثابت توسط ۵ تایی $\{\alpha, \beta, F, G, \phi\}$ نشان داده مشود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عمل ها[11] ی اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $\phi = \{\phi_1, \phi_2, \cdots, \phi_s\}$ مجموعه وضعیت های داخلی اتوماتا، $F: \phi \times \beta \to \phi$ تابع تولید وضعیت جدید اتوماتا و $G: \phi \to \alpha$ تابع خروجی که وضعیت کنونی را به خروجی بعدی می نگارد. اتوماتا های با ساختار ثابت که در این مقاله به انها اشاره شده است در ادامه امده است.

۳-۳-۱- اتوماتای $L_{2N,2}$: این اتوماتا تعداد پاداش ها و جریمه های دریافت شده برای هر عمل را نگهداری کرده و تنها زمانی که تعداد جریمه ها بیشتر از پاداش ها می گردد، عمل دیگر را انتخاب می کند. نمودار تغییر وضعیت این اتوماتا مطابق شکل ۴ می باشد.



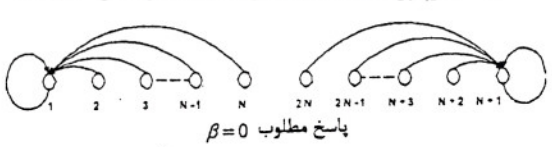
شکل ۴: نمودار تغییر وضعیت اتوماتا $L_{2N,2}$

۳-۳-۲- اتوماتای $G_{2N,2}$: در این اتوماتان بر خلاف $L_{2N,2}$، عمل α_2 حداقل N بار انجام می گردد (پس از گرفتن N جریمه) تا اینکه نهایتا عمل α_1 دوباره انتخاب شود. گراف تغییر وضعیت این اتوماتا برای پاسخ مطلوب مانند اتوماتان $L_{2N,2}$ بوده و برای پاسخ نامطلوب مطابق شکل ۵ می باشد.



شکل ۵: نمودار تغییر وضعیت اتوماتای $G_{2N,2}$

۳-۳-۳- اتوماتای Krinsky: این اتوماتا زمانی که پاسخ محیط نامطلوب است، مانند $L_{2N,2}$ رفتار می کند. اما برای پاسخ مطلوب هر وضعیت $\phi_i (i = 1,2,\cdots,N)$ به وضعیت ϕ_1 و هر وضعیت $\phi_i (i = N+1, N+2, \cdots, 2N)$ به وضعیت ϕ_{N+1} می رود. بنابراین همیشه N پاسخ نامطلوب متوالی لازم است تا اتوماتا عمل خود را عوض کند. نمودار تغییر وضعیت این اتوماتان برای پاسخ نامطلوب مانند اتوماتای $L_{2N,2}$ بوده و برای پاسخ مطلوب مطابق شکل ۶ میباشد.



شکل ۶: نمودار تغییر وضعیت اتوماتای Krinsky

---

[5] Learning Automaton

[6] Environment

[7] Unfavorable

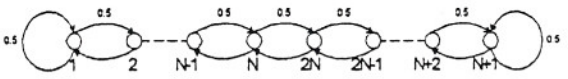
[8] Stationary

[9] Non-Stationary

[10] Fixed Structure

[11] Actions

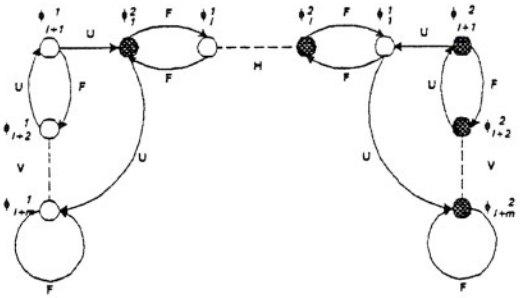
۳-۳-۴- اتوماتای Krylov: در این اتوماتا زمانیکه پاسخ محیط مطلوب است، تغییر وضعیت مانند اتوماتان $L_{2N,2}$ می باشد. اما زمانیکه پاسخ محیط نامطلوب می باشد، هر وضعیت $\phi_i (i \neq 1, N, N+1, 2N)$ با احتمال ۰/۵ به وضعیت ϕ_{i+1} و با احتمال ۰/۵ به وضعیت ϕ_{i-1} مطابق شکل ۷ منتقل می شود:



پاسخ نا مطلوب $\beta = 1$

شکل ۷: نمودار تغییر وضعیت اتوماتا Krylov

۳-۳-۵- اتوماتای Ponomarev: نمودار تغییر وضعیت این اتوماتان مطابق شکل ۸ می باشد. دایره های خالی وضعیت های عمل α_1 و دایره های پر، وضعیت های عمل α_2 را نشان می دهند. در این شکل F به معنای پاسخ مطلوب و U به معنای پاسخ نامطلوب می باشد. شکل دارای دو قسمت عمودی V و یک قسمت افقی H است. قسمت عمودی سمت چپ دارای m وضعیت بوده و تماما مربوط به عمل α_1 می باشد. قسمت عمودی سمت راست نیز دارای m وضعیت بوده و تماما مربوط به عمل α_2 می باشد. در این دو قسمت رفتار اتوماتا شبیه $L_{2N,2}$ است. $2l$ وضعیت افقی در قسمت H بطور یک در میان مربوط به عمل های α_1 و α_2 می باشند. شماره وضعیت های عمل α_1 از راست به چپ بوده و با ϕ_i^1 نشان داده می شوند و شماره وضعیت های عمل α_2 از سمت چپ به راست بوده و با ϕ_i^2 نشان داده می شوند.



شکل ۸: نمودار تغییر وضعیت اتوماتا Ponomarev

وقتی سیستم در یکی از وضعیت های H است، هم پاداش و هم جریمه باعث تغییر عمل می شود. می توان گفت که اتوماتای Ponomarev خصوصیات اتوماتاهای Tsetlin و Krinsky را با هم دارا می باشد. قسمت افقی را پریود آموزش میگوئیم درحالیکه قسمتهای عمودی متناظر با وضعیتهایی هستند که در آنها اتوماتا تصمیم می گیرد تا عملی را که یاد گرفته است، انجام دهد.

۳-۴- اتوماتای یادگیر با ساختار متغیر[۱۲]: اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ تعریف میشود که $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عمل های اتوماتا، $\beta = \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p = \{p_1, p_2, \cdots, p_r\}$ بردار احتمال و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری میباشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله nام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش و سایر احتمالها کاهش می یابند. برای پاسخ نامطلوب $p_i(n)$ کاهش و سایر احتمالها افزایش می یابند. در هر حال، تغییرات بگونه ای صورت می پذیرد که حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک یا قی بماند .الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی میباشد[18].

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)\,p_j(n) \quad \forall j \quad j \neq i$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)\,p_i(n)$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \quad \forall j \quad j \neq i$$

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت زیر را می توان درنظر گرفت:

الف- زمانیکه a و b با هم برابر باشند، الگوریتم را L_{R-P} مینامیم.

ب- زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R-\varepsilon P}$ مینامیم.

پ- زمانیکه b مساوی صفر باشد، الگوریتم را L_{R-I} مینامیم.

برای مطالعه بیشتر درباره اتوماتاهای یادگیرمی توان به مراجع[19-23] مراجعه نمود.

۳- اتوماتای یادگیر سلولی

اتوماتای یادگیر سلولی مدلی برای سیستم هایی است که از اجزاء ساده ای تشکیل شده اند و رفتار هر جزء بر اساس رفتار همسایگانش و نیز تجربیات گذشته اش تعیین و اصلاح می شود. اجزاء ساده تشکیل دهنده این مدل، از طریق تعامل (فعل و انفعال) با یکدیگر می توانند رفتار پیچیده ای را از خود نشان دهند. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول درآن به یک (یا چند) اتوماتای یادگیرمجهز می باشد که وضعیت این سلول را مشخص می کند. مانند اتوماتای سلولی، قانونی محلی Φ بر اتوماتای یادگیر سلولی حاکم است که تعیین می کند که ایا عمل انتخاب شده توسط یک اتوماتا در یک سلول بایستی پاداش داده شود و یا جریمه شود. عمل دادن جریمه ویا پاداش منجر به بروز دراوردن ساختار اتوماتای یادگیر سلولی بمنظور نیل به یک هدف مشخص میگردد.

یک اتوماتای یادگیر سلولی بصورت پنج تایی $< \Lambda, A, \Omega, \Phi, L >$ نشان داده می شود. $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_t\}$ مجموعه سلولهای موجود در اتوماتای یادگیر سلولی می باشد که در یک شبکه کارتزین قرار گرفته‌اند. $A = \{a_1, a_2, \cdots, a_t\}$ مجموعه اعمال مجاز یک اتوماتای یادگیرساکن در یک سلول است. $A'(\lambda_i)$ عمل انجام گرفته در سلول λ_i در زمان t را نشان می دهد و Φ قانون حاکم بر اتوماتای یادگیر سلولی میباشد. $\Omega = \{\Omega_1, \Omega_2, \cdots, \Omega_m\}$ مجموعه سلولهای همسایه یک سلول در اتوماتای یادگیر سلولی بوده و $L=\{L_1, L_2, \ldots, L_n\}$ مجموعه اتوماتاهای یادگیری است که هر سلول به آنها مجهز می باشد. $\Omega(\lambda_i)$ سلولهای همسایه سلول λ_i بوده که دارای دو خصوصیت زیر می باشد:

[۱۲] Variable Structure

1) $\lambda_i \notin \Omega(\lambda_i) \quad \forall \lambda_i \in \Delta$

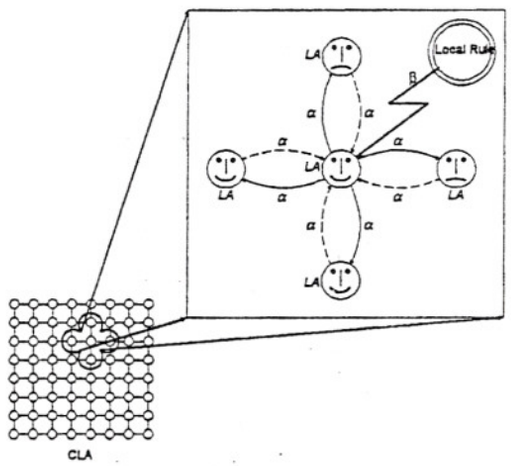
2) $\lambda_i \in \Omega(\lambda_j) \quad \text{iff } \lambda_j \in \Omega(\lambda_i) \quad \forall \lambda_i, \lambda_j \in \Delta$

فرض می کنیم که $W(\lambda_i) = \Omega(\lambda_i) \cup \{\lambda_i\}$ باشد. در آنصورت، قانون حاکم بر سیستم میتواند بصورت تابعی بشکل زیر تعریف شود:

$A^{t+1}(\lambda_i) = \Phi\{A^t(x) | x \in W(\lambda_i)\}$

عملکرد اتوماتای یادگیر سلولی را می توان بصورت زیر شرح داد. در ابتدا هر اتوماتای یادگیر در اتوماتای یادگیر سلولی یکی از اعمال از مجموعه اعمال خود را انتخاب میکند. این عمل می تواند بر اساس مشاهدات قبلی و یا بصورت تصادفی انتخاب شود. عمل انتخاب شده با توجه به اعمال انتخاب شده توسط سلولهای همسایه و قانون حاکم بر اتوماتای یادگیر سلولی پاداش داده میشود و یا جریمه میگردد. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است ساختار داخلی اتوماتا بروز میگردد. عمل بروز در اوردن تمام اتوماتاها در اتوماتای یادگیر سلولی بصورت همزمان انجام میشود. بعد از بروز در اوردن هر اتوماتا در اتوماتای یادگیر سلولی دوباره یک عمل ار مجموعه اعمال خود را انتخاب و انجام میدهد. نتیجه عمل منجر به دادن پاداش و یا جریمه به ان عمل میگردد. فرایند انتخاب عمل و دادن پاداش یا جریمه تا زمانیکه سیستم به یک حالت پایدار برسد و یا یک معیار از قبل تعریف شده ای بر قرار شود ادامه می یابد. عمل بروز در اوردن ساختار اتوماتا های موجود در اتوماتای یادگیر سلولی توسط الگوریتم یادگیری انجام میگیرد.

شکل ۹ اتوماتای یادگیر سلولی را نشان می دهد که در آن از همسایگی von Neumann استفاده شده است. در این شکل، اتوما تاهای هایی که خوشحال هستند در مرحله قبل پاداش و اتوماتاهایی که ناراحت هستند در مرحله قبل جریمه دریافت کرده اند.



شکل ۹: اتوماتای یادگیر سلولی (CLA)

قانون حاکم بر اتوماتای یادگیر سلولی همانند اتوماتای سلولی می تواند بصورت *totalistic* و یا *general* باشد.

ویژگی های اساسی اتوماتای یادگیر سلولی طرح شده در این مقاله عبارتند از: فضا و زمان بصورت گسسته می باشند، هر سلول دارای یک اتوماتای یادگیر میباشد، بروز درآوردن اتوماتاهای یادگیر در سلولها بشکل همگام صورت می پذیرد، اتوماتا های یادگیر تخصیص داده شده به سلولها یکسان می باشند، اتوماتای یادگیر سلولی از همسایگی Moore و با مرز پریودیک استفاده میکند و هر اتوماتای یادگیرداری دو عمل α_1 و α_2 میباشد. مقدار عمل α_1 برابر **یک** و مقدار عمل α_2 برابر **صفر** در نظر گرفته شده است. قوانین بطور قطعی اعمال می شوند و قانون در هر سلول فقط بستگی به اعمال انتخاب شده توسط همسایه های آن سلول دارد.

آنتروپی معیاری است برای اندازه گیری میزان اطلاعاتی که توسط یک منبع تولید می شود و یا توسط فرد شاهد دریافت می گردد.در مدل اتوماتای یادگیرنده سلولی، آنتروپی بعنوان معیاری برای سنجش میزان کارآیی و بررسی سیستم مورد استفاده قرار میگیرد[6].

آنتروپی یک اتوماتای یادگیر LA_{ij} در CLA با مجموعه اعمال $\{\alpha_1, \alpha_2, .., \alpha_r\}$ برابر است با

$$E_{ij} = -\sum_{k=1}^{r} p(\alpha_k) \log(p(\alpha_k))$$

که $p(\alpha_i)$ احتمال آنتروپی های انتخاب عمل α_i توسط اتوماتای یادگیر میباشد. آنتروپی CLA مجموع آنتروپی های اتوماتا های ی یادگیر در CLA مباشد که بصورت زیر تعریف میشود.

$$E = \sum_i \sum_j E_{ij}$$

برای اطلاعات بیشتر درباره اتوماتای یادگیرنده سلولی و کاربردهای آن میتوان به مرجع [24,5] مراجعه نمود.

۵- بررسی نتایج آزمایشات

در این قسمت رفتار اتوماتای یادگیر سلولی برای اتوماتا ها و قوانین مختلف مورد مطالعه قرار گرفته است. در ادامه به برخی از قوانینی که برای آزمایشات استفاده شده است اشاره میشود.

۵-۱- قوانین *General*: برای قوانین *General* نحوه نامگذاری سلولها بر اساس شکل ۱۰ می باشد:

A1	A2	A3
B1	B2	B3
C1	C2	C3

شکل ۱۰: نحوه نامگذاری سلولها

برای قوانین تعریف شده دززیر هر اتوماتا در CLA دارای دو عمل میباشد. اگر اتوماتا عمل α_1 را انتخاب نماید سلول آن اتوماتا بصورت پر ودر غیر اینصورت سلول آن اتوماتا بصورت خالی نشان داده خواهد شد.

۵-۱-۱ قانون *And All*: در این قانون زمانی به عمل انتخاب شده توسط اتوماتا پاداش داده می شود که خود اتوماتا و تمام ۸ همسایه اش عمل α_1 را انتخاب کرده باشند و در غیر اینصورت عمل انتخاب شده توسط اتوماتاجریمه می شود. این قانون بصورت زیربیان میشود.

AND(A1,A2,A3,B1,B2,B3,C1,C2,C3)

با توجه به اینکه ارزش عبارت منطقی فوق ورودی اتوماتا (پاسخ محیط) میباشد. (برای ارزیابی عبارت منطقی فوق سلول پر True و سلول خالی False در نظر گرفته شده است). میتوان قانون فوق را بصورت دیگری نیز بیان کرد. بر اساس قانون فوق اگر یک اتوماتا در CLA عمل α_2 را انتخاب کند ان عمل قطعا جریمه خواهد شدو اگر یک اتوماتا در CLA عمل α_1 را انتخاب کند و همچنین تمام اتوماتاهای همسایه این اتوماتا عمل α_1 را انتخاب کرده باشند عمل انتخاب شده توسط این اتوماتا یعنی عمل α_1 پاداش خواهد گرفت و در غیر این صورت جریمه خواهد شد.

لازم بذکر است که برای قوانین general مقدار True برای یک قانون بعنوان پاسخ مناسب محیط تعبیر میشود.

۵-۱-۲- قانون *Or All*: در این قانون زمانی به عمل انتخاب شده توسط اتوماتا پاداش داده می شود که حداقل اتوماتای مزکزی یا یکی از ۸ اتوماتای همسایه اش عمل α_1 را انتخاب کرده باشند. درغیر اینصورت ان عمل جریمه می شود. این قانون بصورت زیر بیان میشود.

OR(A1,A2,A3,B1,B2,B3,C1,C2,C3)

به بیان دیگر بر اساس قانون فوق اگر یک اتوماتا در CLA عمل α_1 را انتخاب کند ان عمل قطعا پاداش خواهد گرفت .اگر یک اتوماتا در CLA عمل α_2 را انتخاب کند و حداقل یکی از اتوماهای همسایه این اتوماتا عمل α_1 را انتخاب کرده باشد عمل انتخاب شده توسط این اتوماتا پاداش خواهد گرفت و در غیر اینصورت جریمه می شود

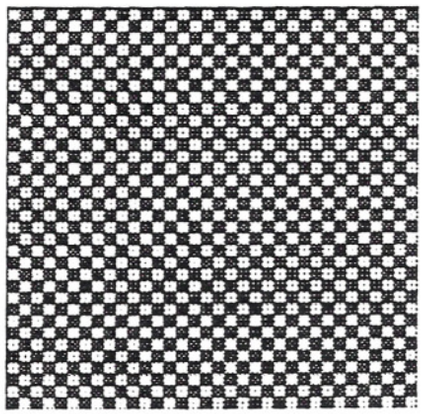
۵-۱-۳- قانون *Cross*: این قانون بصورت زیر بیان میگردد.

OR(A1,A3,B2,C1,C3)

۵-۱-۴- قانون *Homogeneous*: هدف از طرح این قانون آن است که ببینیم آیا اتوماتاها توانایی آن را دارند که با شروع از یک الگوی بی نظم، الگویی را مطابق شکل ۱۱ ایجاد کنند. بدین منظور الگوهایی که باید به آنها پاداش داده شوند مطابق شکل ۱۲ هستند.



شکل ۱۱: الگوهای مطلوب در ساختار نهایی قانون *Homogeneous*

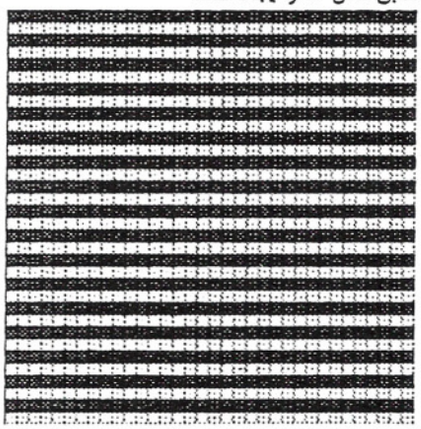


شکل ۱۲: ساختار مطلوب نهایی قانون *Homogeneous*

قانون *Homogeneous* بصورت زیر تعریف میشود.

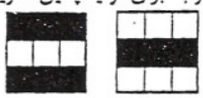
```
OR(
AND(NOT(A1),B1,NOT(C1),A2,NOT(B2),C2,NOT(A3),B3,NOT(C3)),
AND(A1,NOT(B1),C1,NOT(A2),B2,NOT(C2),A3,NOT(B3),C3)
)
```

۵-۱-۵- قانون *Horizontal Lines*: هدف از این قانون آن است که اتوماها در طی زمان یاد بگیرند تا با شروع از یک الگوی بی نظم، الگویی مطابق شکل ۱۳ را ایجاد کنند.



شکل ۱۳: ساختار مطلوب نهایی قانون *Horizontal Lines*

شکل زیر الگوهای مطلوب برای تولیدچنین الگویی را نشان میدهد.

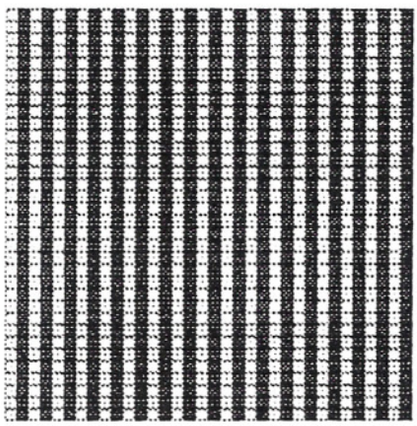


شکل ۱۴: الگوهای مطلوب در ساختار نهایی قانون *Horizontal Lines*

لذا قانون *Horizontal Lines* بصورت زیر تعریف میشود.

```
OR(
AND(A1,B1,C1,NOT(A2),NOT(B2),NOT(C2),A3,B3,C3),
    (NOT(A1),NOT(B1),NOT(C1),A2,B2,C2,NOT(A3),NOT(B3),NOT(
C3))
)
```

۵-۱-۶- قانون *Vertical Lines*: هدف از طرح این قانون ایجاد الگویی بشکل یکسری خطوط موازی و عمودی با فاصله واحد از یکدیگر مطابق شکل ۱۵ می باشد.



شکل ۱۵: ساختار مطلوب نهایی قانون *Vertical Lines*

شکل زیر الگوهای مطلوب برای تولیدچنین الگویی را نشان میدهد.

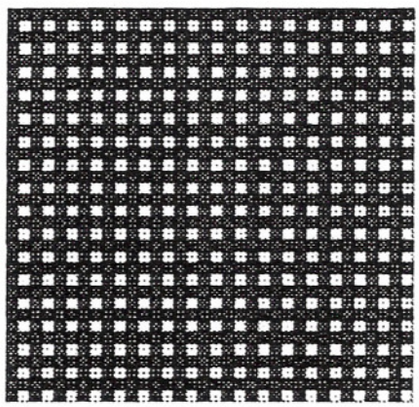


شکل ۱۶: الگوهای مطلوب در ساختار نهایی قانون *Vertical Lines*

قانون *Vertical Lines* بصورت زیر تعریف میشود.

```
OR(
AND(A1,A2,A3,NOT(B1),NOT(B2),NOT(B3),C1,C2,C3),
    (NOT(A1),NOT(A2),NOT(A3),B1,B2,B3,NOT(C1),NOT(C2),NOT(
C3))
)
```

۵-۱-۷- قانون *Grid*: هدف ایجاد الگویی بشکل یک *grid* می باشد.



شکل ۱۷: ساختار مطلوب نهایی قانون *Grid*

با توجه به شکل۱۷ الگوهایی مطلوب عبارتند از:

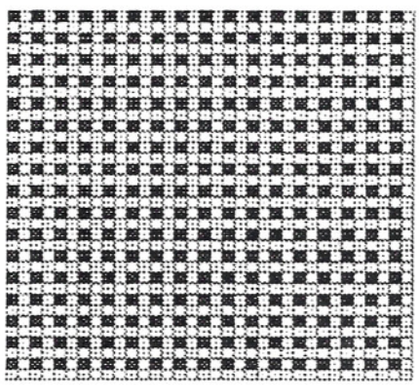


شکل ۱۸: الگوهای مطلوب در ساختار نهایی قانون *Grid*

لذا قانون *Grid* بصورت زیر تعریف میشود.

```
OR(
AND(NOT(A1),B1,NOT(C1),A2,B2,C2,NOT(A3),B3, NOT(C3)),
AND(A1,NOT(B1),C1,A2,B2,C2,A3,NOT(B3),C3),
AND(A1,B1,C1,A2,NOT(B2),C2,A3,B3,C3),
AND(A1,B1,C1,NOT(A2),B2,NOT(C2),A3,B3,C3)
)
```

۵-۱-۸- قانون *Separate*: هدف از این قانون ایجاد الگویی است که در آن هیچ دو اتوماتانی که عمل α_1 را انتخاب کرده باشند در کنار یکدیگر قرار نگیرند.



شکل ۱۹: ساختار مطلوب نهایی قانون *Separate*

برای این قانون الگوهای مطلوب عبارتند از:



شکل ۲۰: الگوهای مطلوب در ساختار نهایی قانون *Separate*

لذا قانون *Separate* بصورت زیر تعریف میشود.

```
OR(
AND(NOT(A1),NOT(B1),NOT(C1),NOT(A2),B2,NOT(C2),NOT(A3),
NOT(B3),NOT(C3)),
AND(A1,NOT(B1),C1,NOT(A2),NOT(B2),NOT(C2),A3,NOT(B3),C3),
    (NOT(A1),B1,NOT(C1),NOT(A2),NOT(B2),NOT(C2),NOT(A3),B3
, NOT(C3)),
    (NOT(A1),NOT(B1),NOT(C1),A2,NOT(B2),C2,NOT(A3),NOT(B3)
, NOT(C3))
)
```

۵-۲- قوانین *Totalistic*: این دسته از قوانین همانند نوع مشابه خود در اتوماتای سلولی میباشند با این تفاوت که از نتیجه قانون برای دادن پاداش یا جریمه استفاده می گردد. این قانون بصورت N-M نشان داده میشود که $M=\{M_1,M_2,...\}$ و $N=\{N_1,N_2,...\}$ میباشد و بصورت زیر تفسیر میگردد.

اگر اتوماتای سلول مرکزی عمل α_1 را انتخاب نماید و تعداد اتوماتاهای همسایه که عمل α_1 را انتخاب کرده باشند متعلق به مجموعه M باشد اتوماتای مرکزی جریمه ودر غیر اینصورت پاداش داده خواهد شد. اگر اتوماتای سلول مرکزی عمل α_2 را انتخاب نماید و تعداد اتوماتاهای همسایه که عمل α_1 را انتخاب کرده باشند متعلق به مجموعه N باشد اتوماتای مرکزی جریمه ودر غیر اینصورت پاداش داده خواهد شد

بعنوان نمونه چند قانون Totalistic در زیر شرح داده میشود.

قانون ۸۹-۰۱۲۳۴۵۶۷: این قانون بصورت زیر میتواند بیان شود. *اگر اتوماتای مرکزی عمل α_1 را انتخاب نماید و تعداد اتوماتاهای همسایه که عمل α_1 را انتخاب کرده باشند بزرگتر یا مساوی ۸ باشد(متعلق به مجموعه M باشد). اتوماتای مرکزی جریمه ودر غیر اینصورت پاداش داده خواهد شد. اگر اتوماتای سلول مرکزی عمل α_2 را انتخاب نماید و تعداد اتوماتاهای همسایه که عمل α_1 را انتخاب کرده باشند کوچکتر یا مساوی ۷ باشد(متعلق به مجموعه N باشد) اتوماتای مرکزی جریمه ودر غیر اینصورت پاداش داده خواهد شد.*

قانون ۹-: این قانون بصورت زیر میتواند بیان شود. *اگر اتوماتای مرکزی عمل α_1 را انتخاب نماید و ۸ اتوماتای همسایه عمل α_1 را انتخاب نمایند به اتوماتای مرکزی جریمه و در غیر اینصورت پاداش داده خواهد شد. اگر اتوماتای مرکزی عمل α_2 را انتخاب نماید پاداش دریافت میکند(این بدین دلیل است که مجموعه N تهی است).*

آزمایشات انجام گرفته شده دراین مقاله با شرایط زیر انجام شده است.

الف- اتوماتای یادگیر با ساختار ثابت: پارامترهای بکار رفته در آزمایشها انجام گرفته با اتوماتای یادگیر با ساختار ثابت بشرح زیر می باشند:

۱. اتوماتای سلولی بکار گرفته شده دارای 35*35 سلول میباشد مگر در مواردی که باید تعداد سطرها و ستونهای اتوماتای سلولی زوج باشد مانند قانون *Minority* با اتوماتای Krylov که بایستی

اتوماتای سلولی با مرز پریودیک باشد که اتوماتای یادگیر سلولی با 34*34 سلول در نظر گرفته شده است.

۲. در برخی از آزمایشات ساختار اولیه تنها شامل یک سلول پر در مرکز اتوماتای یادگیر سلولی می باشد(سلول مرکزی عمل α_1 را انتخاب و بقیه سلولها در CLA عمل α_2 را انتخاب کرده باشد.

۳. در برخی از آزمایشات با اتوماتای Krylov ساختار اولیه با چگالی $\rho_0 = 0$ (همه سلولها سفید هستند) و در برخی ساختار اولیه با چگالی $\rho_0 = 50$ در نظر گرفته شده است

۴. در صورت استفاده از اتوماتا Krylov آزمایش تا رسیدن به ساختار پایدار نهایی ادامه پیدا می کند. برای سایر اتوماتاهاآزمایش تا نسل 2000 ادامه پیدا میکند.

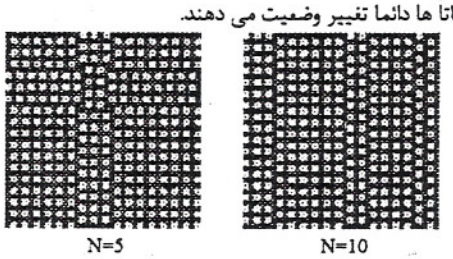
ب- اتوماتای یادگیر با ساختار متغیر: پارامترهای بکار رفته در آزمایشها با اتوماتای یادگیر با ساختار متغیر بدین شرح می باشند:

۱. اتوماتای سلولی بکار گرفته شده دارای 35*35 سلول میباشد مگر در مواردی که باید تعداد سطرها و ستونهای اتوماتای سلولی زوج باشد مانند قانون *Separate* که بایستی اتوماتای سلولی با مرز پریودیک باشد که اتوماتای یادگیر سلولی با 34*34 سلول در نظر گرفته شده است.

۲. احتمال اولیه انتخاب عمل های هر اتوماتا مساوی ۰/۵ در نظر گرفته شده است.

۳. آزمایش زمانی متوقف میشود که یا میانگین آنتروپی به حدود صفر رسیده باشد و یا برای مدتی ثابت باقی مانده باشد.

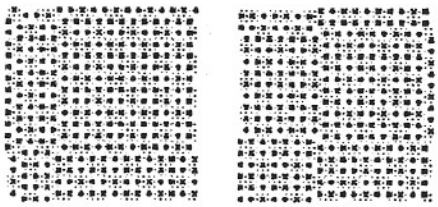
۵-۲-۱- قانون ۸۹-۱۲۳۴۵۶۷: شکل زیر الگوی تولید شده توسط این قانون را نشان میدهد. در صورت استفاده از اتوماتای Krylov و با شروع از حالت اولیه تصادفی با چگالی اولیه $\rho_0 = 50$ الگوهایی ایجاد میشود که در آنها اتوماتاهایی که عمل α_2 را انجام می دهند، در کنار هم قرار ندارند. الگوهای حاصل شباهت بسیار زیادی به الگوی مطلوب در قانون *Grid* دارند. در حالت N=1 به الگوی مطلوب نهایی نمی رسیم و اتوماتا ها دائما تغییر وضعیت می دهند.



N=5 N=10

شکل ۲۱: نتایج حاصل از قانون 89-01234567 و اتوماتان Krylov

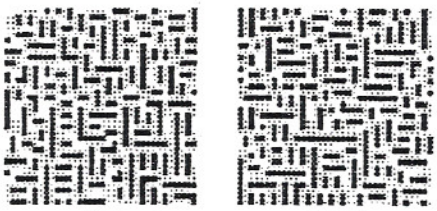
۵-۲-۲- قانون ۸۹-۰۱۲۳۴۵۶۷: در صورت استفاده از اتوماتای Krylov و با شروع از حالت اولیه تصادفی با چگالی اولیه $\rho_0 = 50$ الگوهایی ایجاد میشود که در آنها اتوماتاهایی که عمل α_1 را انجام می دهند، در کنار هم قرار ندارند. الگوهای حاصل شباهت بسیار زیادی به الگوی مطلوب در قانون *Separate* دارند. برای N=1 اتوماتای یادگیر سلولی همگرا نمیشود واتوماتا ها دائما تغییر وضعیت می دهند. شکل ۲۲ برخی از نتایج بدست آمده از این آزمایشها را نشان می دهد.



N=5 N=10

شکل ۲۲: نتایج حاصل از قانون 01-23456789 و اتوماتان Krylov

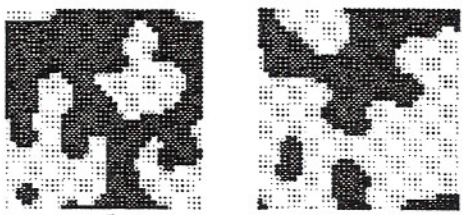
۵-۴-۳- قانون *Cross*: در صورت استفاده از اتوماتای Krylov و با شروع از حالت اولیه با چگالی اولیه $\rho_0 = 0$ الگوهایی شبیه به *maze* ایجاد میشود. در حالت N=1 نیز تا حدودی این الگو ایجاد می شود. این قانون در اتوماتای سلولی الگوی مطلوب در قانون *Homogeneous* را ایجاد می نماید. شکل ۲۳ برخی از نتایج بدست آمده از این آزمایشها را نشان می دهد.



N=5 N=10

شکل ۲۳: نتایج حاصل از قانون *Cross* و اتوماتان Krylov

۵-۴-۴- قانون *Majority* (56789-01234): در صورت استفاده از اتوماتاهای مختلف و با شروع از حالت اولیه تصادفی با چگالی $\rho_0 = 50$ الگوهایی بشکل جزیره ایجاد می شود بطوریکه لبه های تیز در آنها وجود ندارد. در اتوماتاهای با ساختار متغیر با افزایش پارامتر a جزایر کوچکتر شده و تعدادشان افزایش می یابد. این قانون در CA نیز چنین الگویی را ایجاد می نماید. شکل ۲۴ برخی از نتایج حاصل از آزمایشات با این قانون را نشان می دهد. از نتایج این ازمایش در مدلسازی شایعه توسط CLA استفاده شده است[5].

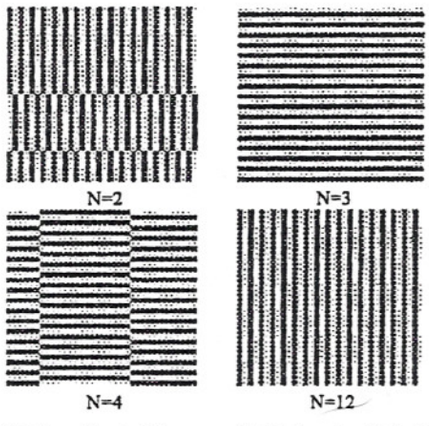


, a=0.01 L_{R-1} $L_{R\&P}$, a=0.01

شکل ۲۴: نتایج حاصل از قانون *Majority*

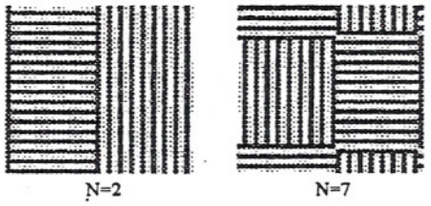
۵-۴-۵- قانون *Minority* (56789-01234): در صورت استفاده از اتوماتای Krylov و با شروع از حالت اولیه تصادفی با چگالی اولیه $\rho_0 = 50$ الگوهایی بصورت یکسری دسته خطوط موازی (عمودی یا افقی) ایجاد می شود. الگوهای حاصل گاهی شبیه به الگوی مطلوب در قانون *Horizontal Lines* و گاهی شبیه به الگوی مطلوب در قانون *Vertical Lines* می باشند. در برخی مواقع نیز شکستگیهایی بصورت جابجایی قسمتی از دسته خطوط مشاهده می شود. چنانچه از

اتوماتای $L_{R-\ell P}$ با پارامتر a=0.90 و محیط با ابعاد زوج استفاده کنیم نیز به این ساختارها می رسیم. در صورتیکه از اتوماتای L_{R-1} استفاده نماییم، به ساختارهایی شبیه به *maze* می رسیم که با افزایش پارامتر a شباهت الگوهای ایجاد شده به الگوی مورد نظر تا حدودی کمتر می شود. بدین صورت که طول کانالها کوتاهتر می گردد. این امر را می توان اینگونه توجیه نمود که زمانیکه پارامتر a افزایش می یابد، سرعت همگرایی اتوماتاها بیشتر شده و لذا احتمال عمل ها سریعتر به مقدار 1 نزدیک میشوند. در نتیجه فرصت کمتری برای یافتن الگوی مطلوب وجود خواهد داشت. در شکل ۲۵ برخی از نتایج حاصل از آزمایشات با قانون *Minority* و اتوماتای Krylov ارائه شده است. از نتایج این آزمایش برای کاربرد CLA در پردازش تصاویر استفاده شده است[5].



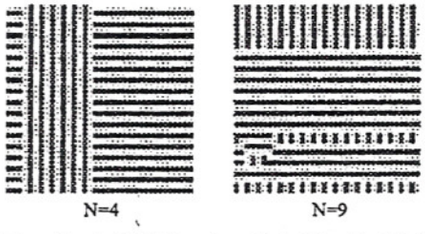
شکل ۲۵: نتایج حاصل از قانون *Minority* و اتوماتای Krylov

۵-۴-۶- قانون *Low Minority* (012345-6789): در صورت استفاده از اتوماتای Krylov و با شروع از حالت اولیه تصادفی با چگالی اولیه $\rho_0 = 50$ الگوهایی بصورت یکسری دسته خطوط موازی (عمودی یا افقی) و یا دو یا چند دسته خطوط موازی و عمود بر هم ایجاد می کنند بطوریکه در نقاط تلاقی، خطوط با هم در تماس هستند. برای حالت N=1 الگوی خاصی ایجاد نمی شود. درصورت استفاده از اتوماتای $L_{R-\ell P}$ با پارامتر a=0.90 و با ابعاد زوج، نیز به این ساختارها می رسیم. در صورت استفاده از اتوماتای L_{R-1} ساختارهایی شبیه به *maze تولید می شود ولی* کانالهای ایجاد شده اکثرا مسدود و با طول کوتاهتر هستند. با افزایش پارامتر a طول کانالها کوتاهتر می شود. در شکل ۲۶ برخی از نتایج حاصل از آزمایشات با قانون *Low Minority* و اتوماتای Krylov آورده شده است.



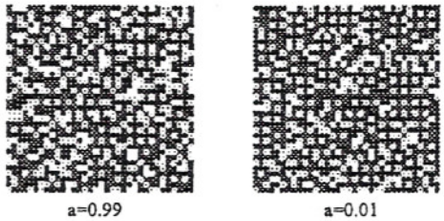
شکل ۲۶: نتایج حاصل از قانون *Low Minority* و اتوماتای Krylov

۵-۴-۷- قانون *High Minority* (0123-456789): در صورت استفاده از اتوماتان Krylov و با شروع از حالت اولیه تصادفی با چگالی اولیه $\rho_0 = 50$ الگوهایی بصورت یکسری دسته خطوط موازی (عمودی یا افقی) و یا دو یا چند دسته خطوط موازی و عمود بر هم ایجاد می کنند بطوریکه در نقاط تلاقی، خطوط با هم در تماس نیستند. برای حالت N=1 الگوی خاصی ایجاد نمی شود. تنها در حالت N=9 در ساختار نهایی الگویی پیش آمد که در آن اتوماتون های واقع در ناحیه مرزی بطور متناوب تغییر حالت می دادند. درصورت استفاده از اتوماتان $L_{R-\ell P}$ با پارامتر a=0.90 و با ابعاد زوج، نیز به این ساختارها می رسیم. در صورت استفاده از اتوماتون L_{R-1} به ساختارهایی که شباهت کامل به *maze* دارند می رسیم که با افزایش پارامتر a طول کانالها در maze کوتاهتر می شود. شکل ۲۷ برخی نتایج حاصل از آزمایشها با قانون *High Minority* و اتوماتان Krylov را نشان میدهد.

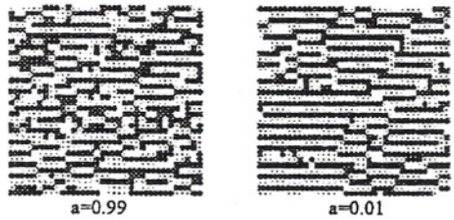


شکل ۲۷: نتایج حاصل از قانون *High Minority* و اتوماتای Krylov

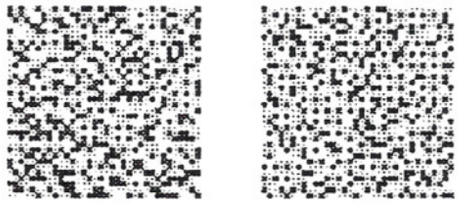
۵-۴-۸- قوانین *Homogeneous* ، *Separate* ، *Grid* ، *Horizontal Lines* و *Vertical Lines*: اشکال زیر الگوی تولید شده توسط قانونهای فوق را نشان میدهد.در صورت استفاده از اتوماتای L_{R1} به ساختارهای مورد نظر نزدیک می شویم. با افزایش پارامتر a (کم شدن عمق حافظه) شباهت ساختارهای ایجاد شده با ساختار مورد نظر کمتر می شود یعنی خطوطی با طول کوتاهتر ایجاد می گردد. این امر را می توان اینگونه توجیه نمود که زمانیکه پارامتر a افزایش می یابد، سرعت همگرایی اتوماتون ها بیشتر شده و در نتیجه فرصت کمتری برای یافتن الگوی مطلوب وجود خواهد داشت. چنانچه از اتوماتای $L_{R\ell P}$ با پارامتر a=0.90 و محیط با ابعاد زوج استفاده کنیم، دقیقا به ساختارهای مورد نظر می رسیم.



شکل ۲۸: نتایج حاصل از قانون *Grid* و اتوماتان L_{R-1}

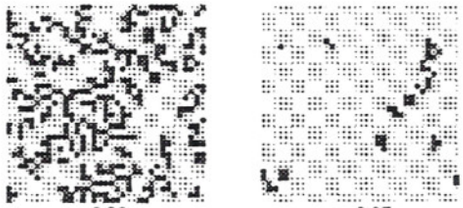


شکل ۲۹: نتایج حاصل از قانون *Horizontal-Lines* و اتوماتان L_{R-I}



شکل ۳۰: نتایج حاصل از قانون *Separate* و اتوماتای L_{R-I}

۵-۴-۹- قانون *Game of Life* (3-01256789): شکل زیر الگوی تولید شده توسط این قانون را نشان میدهد. در صورت استفاده از اتوماتان L_{R-I} با افزایش پارامتر a جمعیت نهایی زیاد می شود. تعداد نسل های سپری شده در a=0.02 تقریبا نصف شده و از آن پس به آرامی و با نوسان کاهش می یابد. در صورت استفاده از اتوماتای L_{R-P} با پارامتر a=0.08 به الگوهایی می رسیم که در ابتدا رشد می کنند و سپس دائما در حال تغییر هستند. در قسمت ۵-۳-۱ عنوان شد که چنانچه از اتوماتای های با ساختار ثابت (بغیر از اتوماتوی Krylov) و با عمق حافظه N=1 استفاده کنیم، اتوماتای یادگیرسلولی رفتارهای اتوماتای سلولی را از خود نشان می دهند. بنابراین، میتوان گفت که اتوماتای یادگیر سلولی دارای ویژگی جهان محاسباتی هستند[8].



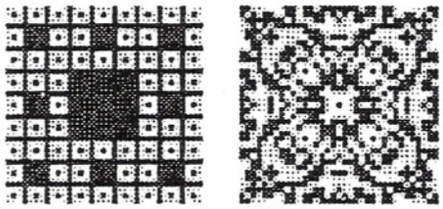
شکل ۳۱: نتایج حاصل از قانون *Game of Life* و اتوماتای L_{RI}

۵-۴-۱۰- قانون *And All*: در صورت استفاده از اتوماتون های L_{RI} و L_{R-P} ، در تمام موارد، محیط به حالتی یکنواخت می رسد بطوریکه تمام اتوماتاهای یا عمل α_1 را انجام می دهند. با افزایش پارامتر a ، نمودار جمعیت خطی موازی محور عمق حافظه می شود. تعداد نسل های سپری شده تا رسیدن به ساختار نهایی نیز بصورت نمایی کاهش می یابد که دلیل این امر بعلت سرعت بیشتر همگرایی اتوماتاهای L_{RI} و L_{R-P} بدلیل افزایش پارامتر a می باشد.

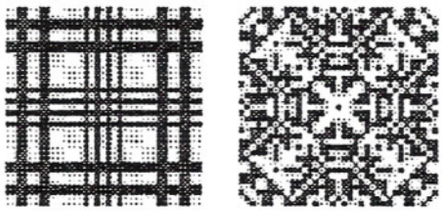
۵-۴-۱۱- قانون *Or All*: در صورت استفاده از اتوماتاهای L_{RI} و L_{R-P} ، در تمام موارد، محیط به حالتی یکنواخت می رسد بطوریکه در هر همسایگی، حداقل یک اتوماتا وجود دارد که عمل α_1 را انجام دهد. با افزایش پارامتر a تعداد نسل های سپری شده تا رسیدن به ساختار نهایی بصورت نمایی کاهش می یابد که دلیل این امر بعلت سریعتر شدن همگرایی اتوماتاهای L_{R-I} و L_{R-P} بر اثر افزایش پارامتر a می باشد.

۵-۴-۱۲- قوانین 02468-02468، 02468-13579، 035789-012469، 0489-269 و 13579-13579: این قوانین بهمراه اتوماتاهای با ساختار ثابت (بغیر از Krylov) در صورت شروع با یک سلول پردر مرکزCLA (سلول مرکزی CLA عمل α_1 وبقیه سلولها عمل α_2 را انتخاب کرده باشند) معمولا ساختارهای زیبایی ایجاد می کنند. نمونه هایی از این ساختارها در شکل های ۳۲ تا ۳۶ آورده شده اند.

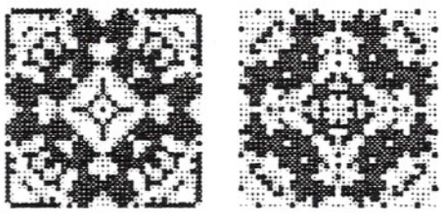
برای بحث مفصلتر در باره CLA و آزمایشات بیشتر به [5] و [24,25] مراجعه کنید.



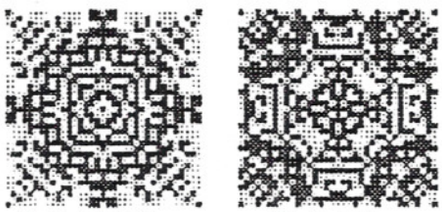
شکل ۳۲: نتایج حاصل از قانون 02468-02468



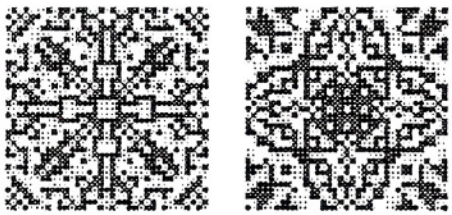
شکل ۳۳: نتایج حاصل از قانون 02468-13579



شکل ۳۴: نتایج حاصل از قانون 035789-012469



شکل ۳۵: نتایج حاصل از قانون 0489-269



Tsetlin-G, N=2 Ponomarev, L=2, M=4

شکل ۴: نتایج حاصل از قانون 13579-13579

۶- نتیجه گیری

با توجه به نتایج بدست آمده در مورد قوانین *general* می توان گفت که اتوماتای $L_{R-\varepsilon P}$ با پارامتر a=0.90 توانایی رسیدن به ساختار مورد نظر را دارد و این امر مستقل از ساختار اولیه آنها می باشد. اتوماتای L_{R-P} و همچنین اتوماتای با ساختار ثابت چنین توانایی را ندارند. اتوماتای L_{R-I} بعلت خاصیت بازگشت ناپذیر آن، تنها در دو مورد یعنی قوانین *And All* و *Or All* چنین توانایی را دارد و در سایر موارد به ساختارهایی نسبتا شبیه به ساختار مورد نظر می رسد. در اتوماتای L_{R-I} هر چه مقدار پارامتر a کوچکتر باشد، ساختارهای بدست آمده به ساختار مطلوب شباهت بیشتری پیدا میکند ولی تعداد نسلهای سپری شده برای رسیدن به ساختار نهایی بیشتر می شود. در مرز دو ناحیه که با هم تطابق ندارند، همیشه تغییر در وضعیت اتوماتاها مشاهده می شود بدون آنکه هیچ یک از گروهها عقب نشینی کنند.

در مورد قوانین *totalistic* ، چنانچه از اتوماتای با ساختار ثابت (بغیر از اتوماتای Krylov) استفاده کنیم، به ساختارهای متقارن و زیبا یی می رسیم. در برخی دیگر از قوانین نظیر انواع قوانین *Minority* چنانچه از اتوماتای Krylov استفاده نماییم، به بعضی از ساختارهای مطلوب در قوانین *general* می رسیم.

با توجه به اینکه اتوماتا های با ساختار ثابت (بغیر از اتوماتای Krylov) و با عمق حافظه مساوی 1 استفاده کنیم، اتوماتای یادگیر سلولی رفتار اتوماتای‌سلولی را از خود نشان می دهند و اینکه در اتوماتون Tsetlin-G افزایش عمق حافظه اثر *delay* را دارد می توان نتیجه گرفت که اتوماتای سلولی حالت خاصی از اتوماتای یادگیرنده سلولی می باشد و همچنین اتوماتای یادگیر سلولی دارای ویژگی جهان محاسباتی میباشد.

مراجع

[1] von Neumann, J., "Theory of Self-Reproducing Automata", University of Illinois Press, 1966.

[2] von Neumann, J., "The General Logical Theory of Automata", Cerebral Mechanisms in Behavior-The Hixon Symposium, 1951.

[3] Liu, Y., "Two Pattern Learning Algorithms Using Dynamical Systems", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1992.

[4] Liu, Y., "Fractals, Neural Networks, Cellular Automata, Formal Language and Coding Theory", Systems, Man and Cybernetics, IEEE International Conference on, Vol. 2, pp. 1663-1669, 1992.

[5] Taherkhani, M., "Proposing and Studying of Cellular Learning Automata as a Tool for Modeling Systems", M.Sc. Thesis, Computer Eng. Dept., Amirkabir University of Technology, Tehran, Iran, 2000.

[6] Wolfram, S., "Statistical Mechanics of Cellular Automata", Review of Modern Physics, Vol. 55, pp. 601-644, 1983.

[7] Wolfram, S., "Twenty Problems in the Theory of Cellular Automata", Physica Scripta, Vol. 9, pp. 170-183, 1985.

[8] Mitchell, M., "Computation in Cellular Automata: A Selected Review", Technical Report, Santa Fe Institute, Santa Fe, U.S.A., 1996.

[9] Adami, C., Introduction to Artificial Life, Springer Verlag, New York, Inc., 1998.

[10] Wolfram, S., "Random Sequence Generation by Cellular Automata", Advances in Applied Mathematics, Vol. 7, pp. 123-169, 1986.

[11] Sasidhar, K., Chattopadhyay, S. and Chaudhuri, P., "CAA Decoder for Cellular Automata Based Byte Error Correcting Code", IEEE Transactions on Computers, Vol. 45, No. 9, 1996.

[12] Srisuchinwong, B., York, T.A. and Tsalides, Ph., "A Symmetric Cipher Using Autonomous And Non-Autonomous Cellular Automata", Global Telecommunications Conference, Vol. 2, pp. 1172-1177, 1995.

[13] Nandi, S., Kar, B.K. and Chaudhuri, P., "Theory and Applications of Cellular Automata in Cryptography", IEEE Transactions on Computers, Vol. 43, No. 12, 1994.

[14] Boccara, N. and Fuks, H., "Modeling Diffusion of Innovations with Probabilistic Cellular Automata", http://www.uic.edu/~boccara/publications.html, 1999.

[15] Sahota, P., Daemi, M.F. and Ellman, D.G., "Training Genetically Evolving Cellular Automata for Image Processing", International Symposium on Speech, Image Processing and Neural Networks, 1994.

[16] Preston, K.JR., Duff, M.J.B., Levialdi, S., Norgren, P.E. and Toriwaki, J.-I., "Basics of Cellular Logic with Some Applications in Medical Image Processing", Proceedings of the IEEE, Vol. 67, No. 5, 1979

[17] Gordillo, J. L. and Luna, J. V., "Parallel Sort on a Linear Array of Cellular Automata", IEEE Int. Conf. on Humans, Information and Technology, 1994.

[18] Narendra, K.S. and Thathachar, M.A.L., Learning Automata: An Introduction, Prentice Hall, Inc., 1989.

[19] Mars, P., Chen, J.R. and Nambir, R., Learning Algorithms: Theory and Applications in Signal Processing, Control and Communications, CRC Press, Inc., 1996.

[20] Lakshmivarahan, S., Learning Algorithms: Theory and Applications, New York, Springer Verlag, 1981.

[21] Meybodi, M.R. and Lakshmivarahan, S., "$\mathcal{E}$ — Optimality of a General Class of Absorbing Barrier Learning Algorithms", Information Sciences, Vol. 28, pp. 1-20, 1982.

[22] Meybodi, M.R. and Lakshmivarahan, S., "On a Class of Learning Algorithms which have a Symmetric Behavior under Success and Failure", Springer Verlag Lecture Notes in Statistics, pp. 145-155, 1984.

[23] Meybodi, M.R., "Results on Strongly Absolutely Expedient Learning Automata", Proc. of OU Inference Conf. 86, ed. D.R. Mootes and R. Butrick, Athens, Ohio: Ohio University Press, pp. 197-204, 1987.

[24] Meybodi, M. R., Beigy, H., and Taherkhani, M., "Cellular Learning Automata And Its Applications", Technical Report, Amirkabir University of Technology, Computer Engineering Department, March 2000.

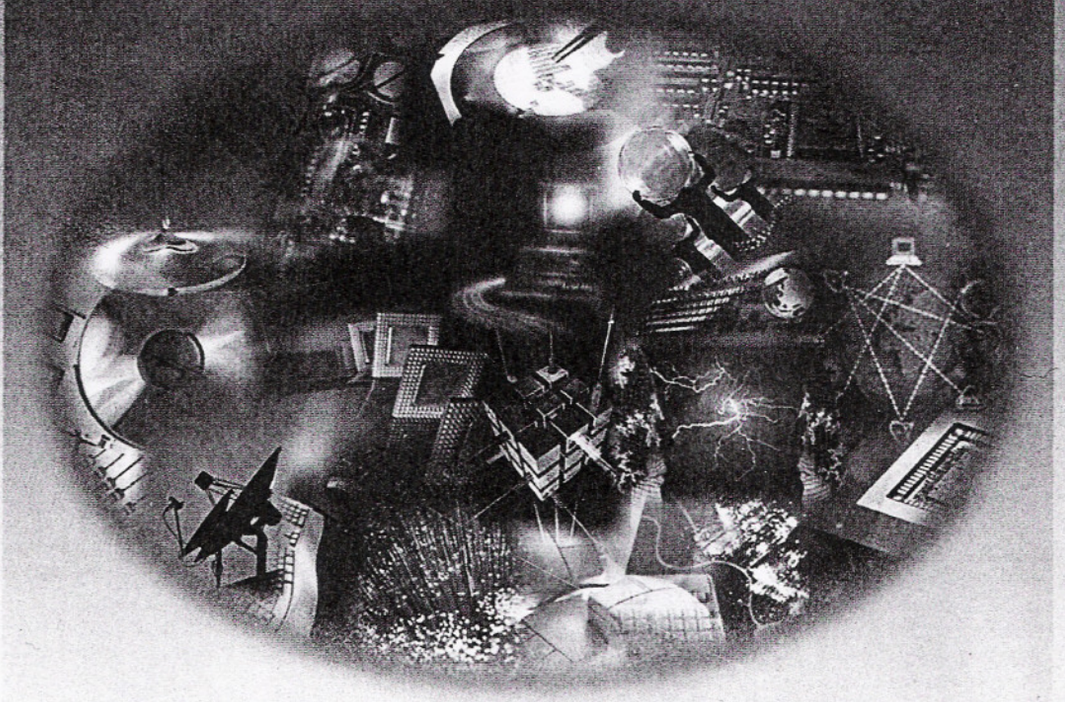
[25] Meybodi, M. R., "Expriments with Cellular Learning Automata", Technical Report, Amirkabir University of Technology, Computer Engineering Department, August 2000.

انجمن کامپیوتر ایران
Computer Society of Iran

IEEE - IRAN



مجموعه مقالات
ششمین کنفرانس بین المللی سالانه
انجمن کامپیوتر ایران
گروه مهندسی کامپیوتر دانشگاه اصفهان



۲ الی ۴ اسفندماه ۱۳۷۹

