سیستم‌های فازی و کاربردها

سال دوم، شماره اول (بهار و تابستان ۱۳۹۸) صص. ۶۹ تا ۸۷

DOR: 20.1001.1.27174409.1398.2.1.4.2

مروری بر اتوماتای فازی*

مرضیه شمسی‌زاده، محمد مهدی زاهدی

دانشگاه تحصیلات تکمیلی صنعتی و فناوری پیشرفته کرمان، بخش ریاضی

چکیده

در این مقاله، در ابتدا ساختار جبری ماشین‌ها و سپس مفاهیم اتوماتای قطعی، غیر قطعی و اتوماتا روی مشبکه‌ی مانده‌ای ارایه شد. همچنین، مفاهیم اتوماتای فازی متناهی و گرامرهای فازی را ارایه داده و ارتباط بین آنها بیان شد. علاوه براین، اتوماتای قطعی و غیر قطعی در حالت کلی با هم مقایسه و دلیل ارایه اتوماتای غیرقطعی بیان گردید.

۱ سرآغاز

اتوماتا در واقع نمونه‌ی اولیه‌ای از سیستم‌های محاسبات عمومی بر روی فضاهای گسسته است و کاربردهای زیادی در زمینه‌های مختلف دارد [۱۵، ۱۳، ۱۲، ۱۱، ۷، ۶، ۴، ۳]. اتوماتا، گرامرها و محاسبه پذیری را در مجموع می‌توان مطالعه توانایی‌های کلی رایانه‌ها تلقی کرد. برای مدل‌سازی سخت‌افزار کامپیوتر، ایده اتوماتا مطرح شده است. اتوماتا ساختاری است که تمام ویژگی‌های لازمه یک کامپیوتر دیجیتالی امروزی را دارا می‌باشد. اتوماتا، ورودی را پذیرفته و خروجی را تولید می‌کند همچنین، ممکن است دارای یک حافظه موقت بوده و تصمیماتی راجع به تبدیل ورودی و خروجی در صورت نیاز اتخاذ نماید.

چرا به مطالعه تئوری اتوماتا می‌پردازیم؟ اولین پاسخ این است که نظریه، مفاهیم و اصولی را

Mathematics Subject Classification (2010): 68Q70, 18B20, 03E72, 06D72 , **Email:** zahedi_mm@kgut.ac.ir .

عبارات و کلمات کلیدی:اتوماتا، اتوماتای فازی، ماشین قطعی، زبان و ماشین‌ها

مطرح می‌کند که در درک ماهیت عمومی رشته کامپیوتر به ما کمک می‌کند. عرصه رشته کامپیوتر شامل دامنه وسیعی از موضوعات خاص از طراحی ماشین تا برنامه‌سازی است. کاربرد کامپیوتر در دنیای واقعی به جزئیات زیادی وابسته است که برای کاربرد و موفقیت آمیزبودن باید آموخته شود.

دومین پیشنهاد و پاسخ روشن این است که ایده‌هایی که بیان و مطرح می‌کنیم کاربردهای مهمی دارند. عرصه‌های طراحی دیجیتال، برنامه‌نویسی و کامپایلرها، مثال‌های ساده ولی بسیار مختلف هستند. اصولی که ما در اینجا مطالعه می‌کنیم مثل نخی بیشتر مفاهیم کامپیوتر، از سیستم عامل تا شناسایی الگوها را به هم پیوند می‌زند.

مجموعه‌ی فازی اولین بار توسط پروفسور لطفی عسکرزاده[1] در سال ۱۹۶۵ ارایه گردید [۱۶]. وی[2] [۱۴] در سال ۱۹۶۷ و سانتوس[3] [۱۰] در سال ۱۹۶۸ ایده‌ای از اتوماتای فازی را ارایه دادند. مفهوم اتوماتای فازی در واقع عمومیت دادن به مفهوم اتوماتای غیرقطعی است. ارتباط بین اتوماتای فازی، اتوماتای غیرقطعی و اتوماتای قطعی توسط ماکور [۸]، بلهولاوک [۱]، لی و پدرسیز [۵] مورد بررسی قرار گرفت. مطالعه اتوماتای فازی روی یک مشبکه‌ی کامل به طور موضعی متناهی، توسط بلهولاوک انجام گرفته است. وی نشان داد هر زبان فازی قابل تشخیص توسط شناسنده‌ی متناهی فازی، می‌تواند توسط یک شناسنده‌ی متناهی قطعی فازی هم تشخیص داده شود.

۲ تعاریف و مفاهیم مقدماتی

تعریف ۱.۲ ([۲]). یک مجموعه‌ی به طور جزیی مرتب، مجموعه‌ای است که در آن رابطه‌ی دوتایی $\leq$ تعریف شده باشد و برای هر x, y, z شرایط زیر برقرار باشند:

۱. انعکاسی باشد یعنی، $x \leq x$.

۲. پادتقارنی باشد یعنی اگر $x \leq y$ و $y \leq x$، آنگاه $x = y$.

[1] Zadeh
[2] Wee
[3] Santos

۳. متعدی باشد یعنی اگر $x \leq y$ و $y \leq z$ باشد، آن‌گاه $x \leq z$.

تعریف ۲.۲ ([۲]). مجموعه‌ی به طور جزیی مرتب L یک مشبکه نامیده می‌شود هرگاه برای هر دو مولفه‌ی موجود در آن کوچکترین کران بالا و بزرگترین کران پایین موجود باشند.

یک مشبکه کامل است اگر هر زیرمجموعه‌ی آن دارای کوچکترین کران بالا و بزرگترین کران پایین باشد.

تعریف ۳.۲ ([۲]). یک مشبکه مانده‌ای جبر $\mathcal{L} = (L, \wedge, \vee, \otimes, \rightarrow, ۰, ۱)$ است که در آن،

۱. $(L, \wedge, \vee, ۰, ۱)$ یک مشبکه است و ۰ کوچکترین عضو آن و ۱ بزرگترین عضو آن می‌باشد.

۲. $(L, \otimes, ۱)$ یک مونوید جابجایی است.

۳. $\otimes$ و $\rightarrow$ دو عمل دوتایی با خاصیت الحاقی به صورت زیر می‌باشد:

$$x \otimes y \leq z \Leftrightarrow x \leq y \rightarrow z.$$

به علاوه، اگر مشبکه‌ی $(L, \wedge, \vee, ۰, ۱)$ یک مشبکه کامل باشد، آن‌گاه مشبکه مانده‌ای $\mathcal{L}$ نیز کامل است.

اعمال $\otimes$ و $\rightarrow$ به ترتیب ضرب و مانده‌ای نامیده می‌شوند.

تعریف ۴.۲. فرض کنید X یک مجموعه ناتهی و متناهی باشد. منظور از X^+، دنباله‌ای به شکل

$$(a_۱), (a_۱, a_۲), \cdots, (a_۱, a_۲, \cdots, a_n), \cdots$$

است که برای هر $i = ۱, ۲, \cdots, n$، $a_i \in A$. حال به منظور سادگی قرارداد می‌کنیم $a_۱ a_۲ \cdots a_n$ نماد $(a_۱, a_۲, \cdots, a_n)$ باشد. در این صورت اگر عمل $\circ$ را به صورت

$$(a_۱ a_۲ \cdots a_l) \circ (b_۱ b_۲ \cdots b_k) = a_۱ \cdots a_l b_۱ \cdots b_k,$$

تعریف کنیم، این عمل شرکت‌پذیر است و X^+ با این عمل، یک نیم‌گروه آزاد است. اکنون با اضافه کردن عضو همانی Λ به X^+ و قرار دادن $X^* = X^+ \cup \{\Lambda\}$ یک مونوید آزاد حاصل می‌شود.

۳ ساختار جبری ماشین‌ها

در ریاضیات، اتوماتا را معادل با یک ماشین و یا حالت خاصی از آن در نظر می‌گیرند. در علوم رایانه، نظریه اتوماتا (ماشین‌ها) عبارت است از مطالعه ماشین‌های محاسبه‌گر و بررسی توانایی آنها برای حل مساله. یک ماشین، ساختاری است که تمام ویژگی‌های یک کامپیوتر دیجیتال را داراست. ماشین، ورودی را دریافت می‌کند خروجی تولید می‌کند. ممکن است انواع مختلفی از ذخیره‌سازی را داشته باشد و می‌تواند در مورد انتقال و تبدیل ورودی به خروجی تصمیم‌گیری نماید.

تعریف ۱.۳. یک ماشین از لحاظ ساختار جبری به صورت $M = (Q, X, Y, f, g, q_\circ)$ می‌باشد به طوری‌که،

۱. Q مجموعه متناهی از حالت‌هاست،

۲. X مجموعه‌ی الفبای ورودی است،

۳. Y مجموعه‌ی الفبای خروجی است،

۴. f تابع انتقال حالت به صورت $f : Q \times X \to Q$،

۵. g تابع خروجی به صورت $g : Q \times X \to Y$،

۶. $q_\circ$ حالت آغازین است.

تعریف ۲.۳. یک ماشین قطعی، یک سه‌تایی به صورت $\mathcal{A} = (Q, X, \delta)$ است که Q و X به ترتیب مجموعه‌ی حالت‌ها و الفبای ورودی نامیده می‌شوند، $\delta : Q \times X \to Q$ یک نگاشت است که تابع انتقال نامیده می‌شود. مجموعه‌ی الفبای ورودی X همیشه متناهی است اما

مجموعه‌ی حالت‌های ممکن است متناهی یا نامتناهی باشد. ماشین قطعی که تعداد حالت‌های آن متناهی باشد ماشین حالت متناهی قطعی [4] نامیده می‌شود.

فرض کنید X^* نمایش یک مونوید آزاد روی X باشد و $\Lambda \in X^*$ کلمه‌ی تهی باشد. در این صورت نگاشت $\delta^* : Q \times X^* \to Q$ توسیع نگاشت δ است و برای هر $a \in Q$ و $u, v \in X^*$ به صورت زیر تعریف می‌شود: $\delta^*(q, \Lambda) = q, \delta^*(q, uv) = \delta(\delta^*(q, u), v)$.

تعریف ۳.۳. فرض کنید X یک الفبای متناهی باشد. هر زیرمجموعه‌ای از X^* را یک زبان و هر عضو از زبان را کلمه گوییم.

فرض کنید X یک الفبای متناهی ناتهی باشد و $\mathcal{L}_۱$ و $\mathcal{L}_۲$ دو زبان زیرمجموعه‌های X^* باشند. آن‌گاه داریم:

$$\mathcal{L}_۱.\mathcal{L}_۲ = \{uv | u \in \mathcal{L}_۱, v \in \mathcal{L}_۲\},$$

و

$$\mathcal{L}_۱^* = \bigcup_{n=۰}^{\infty} \mathcal{L}^n.$$

تعریف ۴.۳. یک اتوماتای قطعی آغازی، یک چهارتایی به صورت $\mathcal{A} = (Q, X, \delta, q_۰)$ است که (Q, X, δ) یک ماشین قطعی است و $q_۰$ یک حالت در Q است که حالت آغازی $\mathcal{A}$ نامیده می‌شود.

مثال ۵.۳. اتوماتای $\mathcal{A}$ را به صورت زیر در نظر بگیرید: $\mathcal{A} = (Q, A, \delta, q_۰, F)$ که در آن $Q = \{q_۰, q_۱, q_۲\}, A = \{a, b\}, F = \{q_۲\}$،

$$\delta(q_۰, a) = q_۱, \qquad \delta(q_۰, b) = q_۱,$$
$$\delta(q_۱, a) = q_۲, \qquad \delta(q_۱, b) = q_۰,$$

اتوماتای $\mathcal{A}$ یک اتوماتای قطعی است.

[4] Deterministic finite automaton

تعریف ۶.۳. یک اتوماتای حالت متناهی غیرقطعی [۵] ، یک پنج‌تایی به صورت

$$\mathcal{A} = (Q, X, \delta, q_0, F),$$

است که در آن

۱. Q مجموعه متناهی از حالت‌هاست،

۲. X مجموعه‌ی ناتهی از الفبای ورودی است،

۳. $\delta : Q \times X \to P(Q)$ تابع انتقال حالت نامیده می‌شود که در آن $P(Q)$ مجموعه توانی Q است،

۴. $q_0 \in Q$ حالت آغازین است،

۵. $F \subseteq Q$ مجموعه حالت‌های نهایی یا پذیرشی است.

مثال ۷.۳. اتوماتای $\mathcal{A}$ را به صورت زیر در نظر بگیرید: $\mathcal{A} = (Q, X, \delta, q_0, F)$ که در آن $Q = \{q_0, q_1, q_2\}, X = \{a, b\}, F = \{q_1\}$ و

$$\begin{aligned}
\delta(q_0, a) &= q_0, & \delta(q_0, a) &= q_2,\\
\delta(q_0, b) &= q_2, & \delta(q_1, a) &= q_0,\\
\delta(q_1, a) &= q_2, & \delta(q_1, b) &= q_0,\\
\delta(q_2, a) &= q_2, & \delta(q_2, b) &= q_2,
\end{aligned}$$

اتوماتای $\mathcal{A}$ یک اتوماتای غیرقطعی است، چون $\delta(q_0, a) = \{q_0, q_2\}$ و $\delta(q_1, a) = \{q_0, q_2\}$.

[5] Nondeterministic finite automaton

تعریف ۸.۳. فرض کنید $\mathcal{A} = (Q, X, \varphi, i, F)$ یک اتوماتا باشد. زبان پذیرش شده توسط $\mathcal{A}$، یک زیرمجموعه از X^* است که به صورت زیر تعریف می‌شود:

$$\mathcal{L}(\mathcal{A}) = \{w \in X^* | \varphi(i, w) \in T\}.$$

مثال ۹.۳. با توجه به تعریف ۸.۳، به وضوح دیده می‌شود که زبان اتوماتای ارایه شده در مثال ۵.۳ برابر $\{a^۲, ba\}$ و زبان اتوماتای مثال ۷.۳ برابر تهی می‌باشد.

تعریف ۱۰.۳. زبان $\mathcal{L}$ را قابل پذیرش گوییم هرگاه اتوماتای مانند $\mathcal{A} = (Q, X, \varphi, i, F)$ وجود داشته باشد به‌طوری‌که $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

تعریف ۱۱.۳. زبان $\mathcal{L}$ را گویا گوییم هرگاه توسط تعداد متناهی اعمال اجتماع، ضرب و ستاره ایجاد شده باشد.

قضیه ۱۲.۳ (قضیه کلین). *فرض کنید X یک الفبای متناهی و $\mathcal{L}$ زیرمجموعه‌ای از X^* باشد. آن‌گاه زبان $\mathcal{L}$ قابل پذیرش است اگر و تنها اگر $\mathcal{L}$ گویا باشد.*

تعریف ۱۳.۳. چهارتایی $\Gamma = (V, A, \pi, \sigma)$ را گرامر گوییم که در آن،

۱. V یک مجموعه‌ی متنهای از الفباست،

۲. A زیرمجموعه‌ی غیرتهی از V است و الفبای نهایی نامیده می‌شوند،

۳. π زیرمجموعه متناهی از $(V - A)^+ \times V^*$ است و قوانین تولید نامیده می‌شود،

۴. $\sigma \in (V - A)$ الفبای آغازین نامیده می‌شود.

و قوانین به صورت $x \to y$ بیان شده و به این شکل عمل می‌کنند که اگر داشته باشیم $w = uxv$ گوییم که قانون $x \to y$ قابل اعمال به رشته $z = uyv$ است و به صورت زیر نوشته می‌شود $w \Rightarrow z$ و گوییم z از w مشتق می‌شود.

تعریف ۱۴.۳. فرض کنید $\Gamma = (V, A, \pi, \sigma)$ یک گرامر باشد. گرامر G را منظم گوییم هرگاه قوانین تولید آن به یکی از دو صورت زیر باشد:

$$\alpha \to x\beta, \alpha \to y,$$

که در آن $\alpha, \beta \in V - A, x \in A^+, y \in A^*$.

مثال ۱۵.۳. گرامر $\Gamma = (V, A, \pi, \sigma)$ را که در آن، $V = \{\alpha, \beta, \sigma, a, b\}, A = \{a, b\}$ و قوانین تولید π به صورت زیر باشد:

$$\sigma \to a\beta, \sigma \to a\alpha, \alpha \to a, \beta \to b,$$

یک گرامر منظم است.

زبان‌های منظم و گرامرهای منظم ارتباط نزدیکی با هم دارند. در واقع این دو مفهوم اساسا یکی هستند، به طوری‌که به ازای هر زبان منظم، یک گرامر منظم و به ازای هر گرامر منظم یک زبان منظم وجود دارد. زبان L را منظم گوییم هرگاه توسط گرامر منظم G پذیرفته شود.

قضیه ۱۶.۳. *هر زبان متناهی منظم است.*

قضیه ۱۷.۳. *فرض کنید X الفبای متناهی و $\mathcal{L}$ زیرمجموعه X^* باشد. زبان $\mathcal{L}$ گویا است اگر و تنها اگر منظم باشد.*

یک ماشین، یک مدل انتزاعی از یک کامپیوتر رقمی است. هر ماشین، شامل برخی ویژگی‌های ضروری است. دارای مکانیسمی برای خواندن ورودی است. فرض می‌شود که ورودی، رشته‌ای روی الفبای داده شده و روی یک فایل ورودی نوشته شده است که ماشین می‌تواند آن را بخواند ولی قادر به تغییر آن نیست. فایل ورودی به سلول‌هایی تقسیم شده است که هریک از آنها قادر به نگهداری یک نماد می‌باشد. مکانیسم ورودی می‌تواند فایل ورودی را از چپ به راست و در هر لحظه یک نماد بخواند. مکانیسم ورودی می‌تواند انتهای رشته ورودی را کشف نماید. ماشین

می‌تواند خروجی را به چندین شکل تولید نماید. ماشین می‌تواند دارای یک دستگاه ذخیره‌سازی موقت باشد که شامل تعداد نامحدودی از سلول‌هاست که هریک قادر به نگهداری یک نماد واحد از الفبا می‌باشند. ماشین می‌تواند محتوای سلول‌های حافظه را خوانده و تغییر دهد. در نهایت، ماشین دارای واحد کنترل است که می‌تواند در هریک از تعداد محدود از حالات داخلی باشد و می‌تواند حالات را به روشی تعریف شده تغییر دهد.

۴ اتوماتای فازی و خواص آنها

تعریف ۱.۴ ([۹]). یک اتوماتای فازی، یک ماشین پنج‌تایی به صورت $\mathcal{A} = (Q, X, \mu, F, \sigma)$ می‌باشد به طوری‌که،

۱. Q مجموعه متناهی غیرتهی از حالت‌هاست،

۲. X مجموعه‌ی متناهی غیرتهی از الفبای ورودی است،

۳. μ تابع انتقال حالت به صورت $\mu : Q \times X \times Q \to [0, 1]$،

۴. F مجموعه حالات نهایی و زیرمجموعه‌ای از Q است،

۵. σ تابع توزیع آغازی به صورت زیر می‌باشد: $\sigma : Q \to [0, 1]$.

مثال ۲.۴. ماشین پنج‌تایی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ ارایه شده به صورت زیر، یک اتوماتای فازی می‌باشد: $Q = \{q_0, q_1, q_2\}$، $X = \{a, b\}$

$$\mu(q_0, a, q_1) = 0/1, \qquad \mu(q_0, b, q_1) = 0/2,$$
$$\mu(q_1, a, q_2) = 0/4, \qquad \mu(q_1, b, q_1) = 0/3,$$

$F = \{q_1\}$ و $\sigma(q_1) = 1$.

تعریف ۳.۴. [۹] اتوماتای فازی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ را قطعی گوییم هرگاه داشته باشیم:

۱. $Im(\mu) \subseteq \{0, 1\}$،

۲. $Im(\sigma) \subseteq \{0, 1\}$،

۳.

$$\Sigma_{q' \in Q}\mu(q, a, q') = 1, \ \ \Sigma_{q' \in Q}\sigma(q') = 1,$$

۴.

$$\mu^*(q, \Lambda, q') = \begin{cases} 1 & \text{اگر} \quad q \neq q' \\ 0 & \text{در غیر این‌صورت} \ quad \end{cases},$$

و

$$\mu^*(q, xa, q') = \Sigma_{q'' \in Q}\mu^*(q, x, q'')\mu^*(q'', a, q),$$

که در آن $q, q', q'' \in Q, x \in X^*, a \in X$.

مثال ۴.۴. اتوماتای فازی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ را که در آن، $Q = \{q_0, q_1, q_2, q_3\}$، $X = \{a, b\}$،

$$\begin{aligned} &\mu(q_0, a, q_1) = 1, && \mu(q_0, b, q_1) = 1, \\ &\mu(q_1, a, q_2) = 1, && \mu(q_1, b, q_1) = 1, \\ &\mu(q_2, a, q_2) = 1, \end{aligned}$$

$F = \{q_1\}$ و $\sigma(q_1) = 1$. اتوماتای $\mathcal{A}$ یک اتوماتای قطعی است.

اتوماتای متناهی غیرقطعی، پیچیده‌تر از انواع قطعی خود هستند. این غیر قطعی بودن، آنها را تبدیل به ابزاری قدرتمند، اما نامتعارف کرده است. ما معمولا کامپیوترها را به صورت کاملا قطعی تصور می‌کنیم. غیرقطعی بودن اتوماتا باعث می‌شود تا بتوان حرکات اتوماتا را انتخاب کرد. به وسیله این ویژگی می‌توانیم به جای اجبار کردن یک حرکت منحصربه‌فرد برای هر وضعیت،

مجموعه‌ای از حرکات مجاز را برای آن پیش‌بینی کنیم. این کار به وسیله تعریف تابع انتقال انجام می‌شود که برد آن مجموعه‌ای از حالات مجاز برای اتوماتاست.

تعریف ۵.۴ ([۹]). اتوماتای فازی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ را غیرقطعی گوییم هرگاه داشته باشیم:

۱. $Im(\mu) \subseteq \{۰, ۱\}$،

۲. $Im(\sigma) \subseteq \{۰, ۱\}$،

۳.

$$\mu^*(q, \Lambda, q') = \begin{cases} ۱ & \text{اگر } q \neq q' \\ ۰ & \text{اگر } O.W \end{cases},$$

و

$$\mu^*(q, xa, q') = \vee_{q'' \in Q} \mu^*(q, x, q'') \wedge \mu^*(q'', a, q),$$

که در آن $q, q', q'' \in Q, x \in X^*, a \in X$.

مثال ۶.۴. اتوماتای فازی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ را که در آن $Q = \{q_۰, q_۱, q_۲, q_۳\}$، $X = \{a, b\}$،

$$\begin{array}{ll} \mu(q_۰, a, q_۱) = ۱, & \mu(q_۰, a, q_۲) = ۱, \\ \mu(q_۱, a, q_۲) = ۱, & \mu(q_۱, b, q_۱) = ۱, \\ \mu(q_۲, a, q_۲) = ۱, & \mu(q_۲, a, q_۱) = ۱, \\ \mu(q_۳, b, q_۲) = ۱, & \end{array}$$

$F = \{q_۱\}$ و $\sigma(q_۱) = ۱$. اتوماتای $\mathcal{A}$ یک اتوماتای غیرقطعی است.

لزوم بررسی غیر قطعی بودن

هنگام استدلال در مورد اتوماتای غیرقطعی، باید با احتیاط بیشتری از مفاهیم شهودی استفاده کنیم. غیرقطعی بودن یکی از مفاهیم پیچیده و مشکل است. کامپیوترهای دیجیتالی کاملا قطعی بوده و حالت آنها در هر زمان منحصرا برحسب ورودی و حالت شروع تعیین می‌شود. سوالی که اینجا مطرح می‌شود این است با توجه به اینکه کامپیوترها کاملا قطعی هستند لزوم مطالعه ماشین‌های غیرقطعی چیست؟ چرا باید روی این ویژگی تمرکز کنیم؟ در بسیاری از الگوریتم‌های قطعی، از قبیل برنامه بازی‌ها، حتما در یکی از مراحل مساله، با مدلی از انتخاب مواجه خواهیم شد. گرچه در اغلب موارد بهترین حرکت نامشخص است، اما می‌توان با جستجو در مسیرهای قبلا پیموده شده، این حرکت را شناسایی کرد. در صورت وجود چندگزینه، یکی را انتخاب و تا زمانی‌که برتری آن اثبات یا رد نشود همان را ادامه می‌دهیم در غیراین‌صورت، به نقطه تصمیم‌گیری قبلی بازگشته و گزینه‌های دیگر را بررسی می‌کنیم. الگوریتم غیرقطعی که بتواند بهترین گزینه را انتخاب نماید قادر است تا بدون بازگشت به مسیر قبلی، حرکت مناسب را شناسایی کند، در حالی‌که الگوریتم قطعی می‌تواند غیرقطعی بودن را شبیه‌سازی کند. به همین دلیل، اتوماتای غیرقطعی را می‌توان به عنوان مدلی برای الگوریتم‌های جستجو و برگشت به عقب مطرح کرد. علاوه براین، یک دلیل فنی برای پرداختن به موضوع غیرقطعی بودن اتوماتا وجود دارد. اثبات برخی نتایج نظری برای اتوماتای غیرقطعی بسیار راحت‌تر از اتوماتای قطعی می‌باشد، هیچ تفاوت اساسی بین این دو نوع اتوماتا وجود ندارد. درنتیجه، مطرح کردن غیرقطعی بودن اتوماتا در اغلب موارد باعث تسهیل روند استدلال‌های دقیق شده و در عین حال هیچ تاثیری بر کلیت نتیجه‌گیری نمی‌گذارد.

تعریف ۷.۴ ([۹]). فرض کنید $\mathcal{A} = (Q, X, \mu, F, \sigma)$ یک اتوماتای فازی باشد. رفتار اتوماتای $\mathcal{A}$ با آستانه c به صورت زیر تعریف می‌شود:

$$\mathcal{B}(\mathcal{A}, c) = \{x \in X^* | \vee_{p\in Q, q\in F}\, \sigma(p) \wedge \mu^*(p, x, q) > c\}.$$

قضیه ۸.۴ ([۹]). *فرض کنید* $\mathcal{A} = (Q, X, \mu, F, \sigma)$ *یک اتوماتای فازی باشد.* $\mathcal{B}(\mathcal{A}, c) \neq \emptyset$ *اگر و فقط اگر* $x \in \mathcal{B}(\mathcal{A}, c)$ *به‌طوری‌که* $|x| \leq |Q|$.

قضیه ۹.۴ ([۹]). *برای هر اتوماتای فازی* $\mathcal{A}$ *و* $0 \leq c < 1$ *، یک اتوماتای غیر قطعی فازی*

مانند $\mathcal{A}^c$ وجود دارد به طوری‌که $\mathcal{B}(\mathcal{A}, c) = \mathcal{B}(\mathcal{A}^c, \circ)$.

قضیه ۱۰.۴ ([۹]). *برای هر اتوماتای فازی $\mathcal{A}$ و $\circ \leq c < ۱$ ، یک اتوماتای قطعی فازی مانند $\mathcal{A}'$ وجود دارد به طوری‌که* $\mathcal{B}(\mathcal{A}, c) = \mathcal{B}(\mathcal{A}', \circ)$.

در منطق ریاضی، به زبان‌هایی که با فرمول‌های دقیق قابل پردازش برای ماشین، تعریف شده باشند زبان‌های نرمال یا زبان‌های صوری گفته می‌شود. زبان صوری، مدل انتزاعی از ویژگی‌های عمومی زبان‌های برنامه‌سازی است. بررسی زبان‌های صوری امکانی را برای یادگیری زبان‌های برنامه‌سازی در اختیار خواننده قرار می‌دهد. زبان‌های صوری زبان‌هایی هستند که توسط گرامرها تولید می‌شوند یا ماشین، برای ارزیابی آن‌ها وجود دارد.

تعریف ۱۱.۴. یک زبان فازی از X^*، یک زیرمجموعه‌ی فازی از X^* است. یک اتوماتای فازی $\mathcal{A} = (Q, X, \mu, F, \sigma)$ زبان فازی $f \in \mathcal{F}(X^*)$ را تشخیص می‌دهد اگر برای هر $u \in X^*$ داشته باشیم:

$$f(u) = \vee_{p \in Q} \vee_{q \in F} \sigma(q) \otimes \mu^*(q, u, p),$$

و این به این معنی است که مقدار عضویت کلمه‌ی u در زبان فازی f معادل است با درجه‌ای که با آن، اتوماتای فازی $\mathcal{A}$ کلمه‌ی u را تشخیص می‌دهد.

۵ گرامرها و زبان‌ها

کاربرد گرامرها، غالبا روشی برای تشخیص زبان‌ها محسوب می‌شود. هروقت که با استفاده از اتوماتا یا به روشی دیگر اقدام به تعریف خانواده یک زبان می‌کنیم، بهتر است از گرامر مرتبط با آن خانواده هم اطلاع داشته باشیم.

تعریف ۱.۵ ([۹]). یک گرامر فازی یک چهارتایی به صورت $\Gamma = (T, N, P, h)$ است که در آن،

۱. T و N مجموعه‌های مجزای ناتهی متناهی هستند،

۲. P یک مجموعه متناهی از قوانین تولید فازی روی $T \cup N$ است به قسمی‌که، برای هر $\rho \in P$ و $s \in (T \cup N)^* N (T \cup N)^*$ داریم: $\rho(s,t) > 0$،

۳. h یک تابعی است از N به $[0,1]$.

در این تعریف T و N به ترتیب مجموعه‌های پایانی و غیرپایانی نامیده می‌شوند. $h(A)$ درجه عضویت الفبای آغازین A از گرامر Γ را مشخص می‌کند.

مثال ۲.۵. چهارتایی $\Gamma = (T, N, P, h)$ را که در آن $T = \{a, b\}$، $N = \{\alpha, \beta\}$، $\rho(\alpha, a) = 0/3$، $\rho(\beta, b) = 0/5$، $h(\alpha) = 1$ و $h(\beta) = 0/2$ یک گرامر فازی است.

یک زبان در صورتی منظم است که توسط اتوماتای متناهی پذیرفته شود، بنابراین، هر زبان منظمی را می‌توان به‌وسیله یک اتوماتای قطعی یا غیرقطعی تعریف کرد. به طور نمونه از این توصیف بسیار مفید و کارآمد می‌توان برای نمایش منطق تصمیم‌گیری در مورد وجود یا عدم وجود رشته‌ای مفروض در یک زبان خاص استفاده کرد. اما در بسیاری موارد، این توصیف کافی نبوده و باید روش‌های دقیق‌تری برای توصیف زبان‌های منظم در اختیار داشته باشیم.

تعریف ۳.۵. فرض کنید $\Gamma = (T, N, P, h)$ یک گرامر فازی باشد. آن‌گاه

۱. G یک گرامر مستقل از متن است اگر برای هر $\rho \in P$ و $s, t \in (T \cup N)^*$، $\rho(s,t) > 0$ نتیجه دهد $|s| \leq |t|$.

۲. G یک گرامر منظم است اگر برای هر $\rho \in P$، $s, t \in (T \cup N)^*$ و $\rho(s,t) > 0$ نتیجه دهد $s \in N, t \in aA$ که در آن $a \in T$ و $A \in N \cup \{\Lambda\}$.

مثال ۴.۵. گرامر ارایه شده در مثال ۲.۵، یک گرامر منظم و مستقل از متن می‌باشد.

تعریف ۵.۵ ([۹]). یک زبان فازی λ روی S، یک تابع از S^* به $R^{\geq 0}$ می‌باشد. برای $s \in S^*$، $\lambda(s)$ درجه عضویت s به عنوان عضوی از زبان می‌باشد. اگر $\Gamma = (T, N, P, h)$ یک گرامر باشد، پس λ_G زبان فازی روی T تعریف می‌شود.

قضیه زیر ارتباط بین اتوماتای فازی و گرامرهای فازی را نشان می‌دهد.

قضیه ۶.۵ ([۹]). λ *یک زبان فازی منظم است اگر و فقط اگر یک اتوماتای فازی مانند* M *وجود داشته باشد به‌طوری‌که* λ *زبان اتوماتای فازی* M *باشد.*

۶ اتوماتای فازی روی مشبکه مانده‌ای

در این بخش با توجه به مفهوم مشبکه مانده‌ای، تعریف اتوماتای فازی در فضای مشبکه مانده‌ای را ارایه می‌دهیم.

تعریف ۱.۶. یک ماشین فازی روی مشبکه مانده‌ای $\mathcal{L} = (L, \wedge, \vee, \otimes, ۰, ۱)$، یک سه تایی به شکل $\mathcal{A} = (Q, X, \delta)$ است به طوری‌که Q مجموعه‌ی حالات، X مجموعه غیرتهی الفبای ورودی و δ یک زیرمجموعه‌ی فازی از $Q \times X \times Q$ است و داریم: $\delta : Q \times X \times Q \to L$، که آن را تابع انتقال فازی می‌نامیم.

در واقع $\delta(q, x, p)$ درجه‌ای است که تحت آن کلمه‌ی x از q به p منتقل می‌شود. الفبای X همواره متناهی است. اگر در ماشین فازی مجموعه‌ی حالات هم متناهی باشد، آنگاه آن را ماشین فازی متناهی می‌نامیم.

حال فرض کنید X^*، روی الفبای X یک مونوید آزاد باشد و $\Lambda \in X^*$ عضو همانی آن باشد. آنگاه تابع δ می‌تواند به تابع δ^* توسعه پیدا کند. درنتیجه، برای هر $a \in X, x \in X^*$ و $p, q \in Q$ داریم: $\delta^* : Q \times X^* \times Q \to L$

$$\delta^*(q, \Lambda, p) = \begin{cases} ۱ & \text{اگر} quad p = q \\ ۰ & \text{در غیر این صورت} \end{cases}, \qquad (۱)$$

و داریم:

$$\delta^*(q, xa, p) = \vee_{r \in Q} \delta^*(q, x, r) \otimes \delta(r, a, p).$$

تعریف ۲.۶. ماشین فازی معکوس از ماشین فازی $\mathcal{A} = (Q, X, \delta)$، یک ماشین فازی $\bar{\mathcal{A}} = (Q, X, \bar{\delta})$ است که برای هر $q, p \in Q$ و $x \in X$ تابع انتقال فازی آن به صورت زیر

تعریف می‌شود:

$$\bar{\delta}(q, x, p) = \delta(p, x, q) \tag{۲}$$

به طور کلی اتوماتای فازی معکوس از اتوماتای فازی $\mathcal{A} = (Q, X, \delta, \sigma, \tau)$، اتوماتای فازی $\bar{\mathcal{A}} = (Q, X, \bar{\delta}, \bar{\sigma}, \bar{\tau})$ است که تابع انتقال فازی $\bar{\delta}$، مشابه (۲) و مجموعه‌های فازی از حالات آغازین و نهایی به صورت زیر تعریف می‌شود: $\bar{\sigma} = \tau, \bar{\tau} = \sigma$.

تعریف ۳.۶. یک زبان فازی از X^* روی یک مشبکه مانده‌ای به شکل

$$\mathcal{L} = (L, \wedge, \vee, \otimes, \to, ۰, ۱),$$

یک زیرمجموعه‌ی فازی از X^* است. یک اتوماتای فازی $\mathcal{A} = (Q, X, \delta, \sigma, \tau)$ زبان فازی $f \in \mathcal{F}(X^*)$ را تشخیص می‌دهد اگر برای هر $u \in X^*$ داشته باشیم:

$$f(u) = \vee_{p,q \in Q} \sigma(q) \otimes \delta^*(q, u, p) \otimes \tau(p),$$

و این به این معنی است که مقدار عضویت کلمه‌ی u در زبان فازی f معادل است با درجه‌ای که با آن، اتوماتای فازی $\mathcal{A}$ کلمه‌ی u را تشخیص می‌دهد.

در اتوماتای فازی $\mathcal{A} = (Q, X, \delta, \sigma, \tau)$ اگر $\delta : Q \times X \times Q \to \{۰, ۱\}$، آن‌گاه $\mathcal{A}$ یک اتوماتای صریح غیر قطعی معمولی است.

قضیه ۴.۶ ([۵]). *فرض کنید $\mathcal{A} = (Q, X, \delta, \sigma, \tau)$ یک اتوماتای متناهی قطعی روی مشبکه مانده‌ای $\mathcal{L} = (L, \wedge, \vee, \otimes, \to, ۰, ۱)$ باشد، آن‌گاه برد زبان فازی قابل تشخیص توسط این اتوماتا، زیرمجموعه‌ای متناهی از L است.*

قضیه ۵.۶ ([۵]). *برای زبان فازی قابل تشخیص توسط یک اتوماتای فازی قطعی، اتوماتای فازی غیرقطعی وجود دارد به طوری‌که این زبان را تشخیص می‌دهد، اما عکس این مطلب برقرار نیست.*

۷ برخی کاربردها

اگرچه ما روی خلاصه و قسمت‌های ریاضی از زبان‌های صوری و ماشین‌ها تاکید می‌کنیم، این‌گونه برداشت می‌شود که این مفاهیم کاربرد گسترده‌ای در علم کامپیوتر دارند. درحقیقت یک موضوع معمولی است که به همه قسمت‌های خاص مرتبط می‌شود.

زبان‌های صوری و گرامرها در ارتباط با زبان‌های برنامه‌نویسی کاربرد زیادی دارند. در بیشتر برنامه‌نویسی‌ها ما با درک شهودی کم یا زیاد درباره زمانی که می‌نویسیم کار می‌کنیم، اگرچه ممکن است در آینده دور از آنها استفاده کنیم. ما نیاز داریم به شرح دقیقی از جدول دستورات رجوع کنیم تا بیشتر متن‌های برنامه را بفهمیم. اگرچه ما یک کامپایلر یا برنامه تصحیح‌کننده را بنویسیم، یک شرح دقیق از زبان‌های مورد نیاز در هر مرحله را داریم. در این میان، راهی که زبان‌های برنامه نویسی می‌توانند تعریف شوند، گرامرهایی هستند که به طور گسترده مورد استفاده قرار می‌گیرند.

برای زبان‌های برنامه‌نویسی، کامپایلر نوشته می‌شود. کامپایلر نیاز به تعریف دقیق و رسمی آن زبان برنامه‌نویسی دارد. لذا، می‌توان در زبان‌های برنامه‌نویسی، از گرامر یا اتوماتا برای پذیرش یا عدم پذیرش یک قطعه کد توسط آن زبان برنامه‌نویسی استفاده نمود.

کاربرد دیگر اتوماتا در طراحی رقمی می‌باشد. اتوماتا در طراحی رقمی، توصیف علمی بسیار سطح بالا از یک مدار منطقی و پیاده‌سازی منطقی آن با تراتریستور، گیت و فلیپ فلاپ است.

۸ نتیجه‌گیری

در این مقاله، در ابتدا ساختمان جبری ماشین ها و سپس مفاهیم اتوماتای قطعی، غیر قطعی و گرامرهای فازی بیان گردید. در ادامه، به بررسی علت ارایه اتوماتای قطعی پرداخته شد.

مراجع

[1] Belohlavek, R. (2002) Determinism and fuzzy automata, *Information Sciences*, 143, 205–202.

[2] Belohlávek, R. (2002)*Fuzzy Relational Systems: Foundations and Principles, Kluwer,* NewYork.

[3] Cattaneo, G. Flocchini, G. Mauri, G. Vogliotti, C. Q. Santoro, N. (1997) Cellular automata in fuzzy backgrounds, *Physica D: Nonlinear Phenomena* 105, 105-120.

[4] Doostfatemeh, M. Kremer, S. C. (2003) *A fuzzy finite-state automaton that unifies a number of other popular computational paradigms*, Proceedings of the ANNIE 2003 Conference (ANNIE 03Õ), ASME Press, New York.

[5] Li, Y. M. (2008) Approximation and robustness of fuzzy finite automata, *International Journal of Approximate Reasoning*, 47, 247-257.

[6] Li, Y. Pedrycz, W. (2005) Fuzzy finite automata and fuzzy regular expressions with membership values in lattice-ordered monoids, *Fuzzy sets and system*s, 156, 68-92.

[7] Li, Y. Shi, Z. K. (2000) Remarks on uninorm aggregation operators, *Fuzzy Sets and Systems*, 114, 377-380.

[8] Lin, F. Ying, H. (2002) Modeling and control of fuzzy discrete event systems, *IEEE Transaction on Systems, Man and Cybernetics B*, 32, 401–415.

[9] Mordeson, J. N. Malik, D. S. (2002) Fuzzy automata and languages: theory and applications, *Chapman and Hall/CRC.*

[10] Santos, E. S. (1968) Maximin automata, *Information and Control*, 13, 363-377.

[11] Shamsizadeh, M. Zahedi, M. M. (2016) A note on ”Quotient structures of intuitionistic fuzzy finite state machines, *Journal of Applied Mathematics and Computing*, 51, 413-423.

[12] Shamsizadeh, M. Zahedi, M. M. (2016) Minimal and statewise minimal intuitionistic general L-fuzzy automata, *Iranian Journal of Fuzzy Systems*, 13, 131-152.

[13] Shamsizadeh, M. Zahedi, M. M. Abolpour, Kh. (2016) Bisimulation for BL-general fuzzy automata, *Iranian Journal of Fuzzy Systems*, 13, 35-50.

[14] Wee, W. G. (1968) On generalizations of adaptive algorithms and application of the fuzzy sets concept to pattern classification, 4587-4587.

[15] Ying, M. (2002) A formal model of computing with words, Fuzzy Systems, *IEEE Transactions on*, 10, 640-652.

[16] Zadeh, L. A. (1965) Fuzzy sets, *Information and control*, 8, 338-353.