ریاضی و جامعه
شاپا (چاپی): ۶۴۹۳-۲۳۴۵، شاپا(الکترونیکی): ۶۵۰۷-۲۳۴۵
جلد ۵، شماره ۱ (۱۳۹۹)، صص. ۳۵-۱۷

http://math-sci.ui.ac.ir
http://www.ui.ac.ir

دسته‌بندی چامسکی در نظریه محاسبه

سمیه تاری

چکیده. آشنایی کامل با مفاهیم اساسی و پایه ای در هر زمینه علمی، برای پیشرفت های آن زمینه، لازم و ضروری است. رشد روزافزون مباحث علوم کامپیوتر چه از دیدگاه توصیف مکانیکی، چه از دیدگاه توصیف صوری، نیز این مطلب را می‌طلبد. بنابراین در این مقاله به یکی از مباحث اساسی در زمینه نظریه محاسبه (دسته‌بندی چامسکی) به طور خلاصه پرداخته شده است.

۱. مقدمه

عملکرد مغز انسان در یادگیری و انجام فعالیت های مختلف از دیرباز مورد توجه بشر بوده است. مدل های انتزاعی از آن، که بخشی از فعالیتهای مغز را شبیه سازی کند، در حوزه‌های مختلف توسعه یافته است. مطالعات فراوانی در حوزه علوم کامپیوتر در این جهت انجام گرفته است که در جهت توانمندسازی کامپیوترها پیشرفت های مختلفی داشته است که منجر به برخورداری هر چه بیشتر از توانمندهای مغز در این نوع ابزار شده است. از مفهوم تخصصی زبان در مباحث صوری علوم کامپیوتر استفاده می‌شود. به هر مجموعه متناهی از نمادها یک الفبا[1] گفته می‌شود. فرض کنید Σ یک الفبا باشد. هر دنباله متناهی از نمادهای موجود در Σ یک رشته[2] نامیده می‌شود. رشته پوچ یا تهی، که با نماد λ نشان داده می‌شود، رشته‌ای است که در آن هیچ نمادی از Σ به کار نرفته است. مجموعه تمام رشته‌های متشکل از اعضای Σ با نماد Σ^* نشان داده می‌شود. هر زیر مجموعه از Σ^* یک زبان [3] نامیده می‌شود.

یك ماشین (اتوماتون) را در ساده ترین حالت می‌توان تابعی در نظر گرفت که یك ورودی را دریافت می کند، برحسب ضابطه خود عملیاتی را روی آن ورودی انجام می‌دهد و در پایان یك خروجی را ایجاد می کند. بنابراین می‌توان یك اتوماتون را یك مدل انتزاعی از کامپیوتر دانست. در دهه‌ی ۱۹۳۰ تورینگ[4] ماشینی را تحلیل و بررسی کرد که توانایی‌های کامپیوترهای امروزی را داشت (امروزه مدل مورد بررسی به ماشین تورینگ معروف است). در واقع هدف او این بود که بین کارهایی که ماشین قادر به انجام آن‌ها بود و کارهایی که ماشین قادر به انجام آن‌ها نبود، یک مرزبندی را ایجاد کند. در دهه‌های ۱۹۴۰ و ۱۹۵۰ انواع ساده‌تری از مدل‌های محاسبه که امروزه به اتوماتای متناهی

عبارات و کلمات کلیدی: اتوماتون متناهی قطعی و غیرقطعی، اتوماتون پشته‌ای، ماشین تورینگ، گرامر منظم، گرامر حساس به متن، گرامر نامقید، گرامر مستقل از متن
نوع مقاله: ترویجی
دبیرتخصصی رابط: صغری نوبختیان
تاریخ دریافت: ۱۳۹۹/۰۷/۲۷ تاریخ پذیرش: ۱۳۹۹/۱۰/۲۷
http://dx.doi.org/10.22108/msci.2021.125480.1393

.

[1]Alphabet [2]String [3]Language [4]Turing

معروف شده‌اند، توسط تیمی متشکل از ریاضیدانان، زیست‌شناسان، روانشناسان، مهندسان و متخصصین اولیه نظریه علوم کامپیوتر بررسی شد [۲].

از طرف دیگر در اواخر دهه ۱۹۵۰ زبان‌شناس معروف چامسکی[۵]، مطالعه صوری زبان‌ها را آغاز کرد. وی زبان‌ها را به چهار دسته نوع ۰، نوع ۱، نوع ۲ و نوع ۳ تقسیم‌بندی کرد که به دسته‌بندی چامسکی معروف شده است [۴].

چامسکی از گرامرها برای توصیف زبان‌ها استفاده کرده است که آن توصیف صوری زبان نامیده می‌شود. از طرف دیگر از اتوماتا نیز می‌توان برای توصیف زبان‌ها استفاده کرد که توصیف مکانیکی خوانده می‌شود. در ادامه برای هر دسته از زبان‌ها در بخشهای جداگانه هر دو توصیف و ارتباط بین آن‌ها مطالعه و مرور می‌شوند.

۲. زبان‌های نوع ۳ از دسته‌بندی چامسکی

در این بخش تعریف مکانیکی و صوری زبان‌های نوع ۳، دسته‌بندی چامسکی آورده می‌شود [۳، ۴، ۶]. در توصیف صوری آن‌ها از مفهوم گرامر منظم و در توصیف مکانیکی آن‌ها از مفهوم اتوماتون متناهی استفاده می‌شود.

هر گرامر چگونگی تولید رشته‌های موجود در یک زبان را توصیف می‌کند، که با جایگزینی متوالی نمادها این کار انجام می‌شود.

تعریف ۱.۲. *یک گرامر خطی راست[۶] (به ترتیب خطی چپ[۷])، چهارتایی $G=(V,\Sigma,S,P)$ است که در آن:*

- *V یک مجموعه متناهی است که آن را مجموعه متغیرها می‌نامند.*
- *Σ یک الفبا است. اعضای Σ را ترمینال گویند.*
- *$S\in V$ متغیر شروع است.*
- *P مجموعه قواعد G است که هر قاعده از آن به فرم $A\to\alpha$ یا $A\to\alpha B$ (به ترتیب به فرم $A\to B\alpha$ یا $A\to\alpha$) است که $A,B\in V$ و $\alpha\in\Sigma^*$.*

هر گرامر خطی راست یا خطی چپ یك گرامر منظم[۸] نامیده می‌شود.

مثال ۲.۲. *الف) $G=(\{S\},\{a,b\},S,P=\{S\to Sa, S\to Sb, S\to\lambda\})$ یک گرامر منظم است.*

ب) $G=(\{S,X\},\{a,b\},S,P=\{S\to aS, S\to bX, S\to\lambda, X\to aS\})$ یک گرامر منظم است.

تبصره ۳.۲. *برای سادگی، هر گرامر تنها با مجموعه قواعد آن مشخص می‌شود. به ازای هر متغیر $X\in V$، به جای مشخص کردن قواعد متناظر با X به صورت مجزا، همه قواعد در یک سطر با استفاده از جداکننده "$|$" مشخص می‌شود.*

در مثال ۲.۲ قسمت (الف) مجموعه قواعد متناظر با متغیر S با $S\to Sa\mid Sb\mid\lambda$ مشخص می‌شود و گرامر G به صورت $G: S\to aSa\mid bSb\mid\lambda$ نمایش داده می‌شود.

همچنین در قسمت (ب) قواعد متناظر با متغیر S به صورت $S\to aS\mid bX\mid\lambda$ و قواعد متناظر با متغیر X به صورت $X\to aS$ است. گرامر G به صورت زیر مشخص می‌شود:

$$G:\begin{cases} S\to aS\mid bX\mid\lambda\\ X\to aS\end{cases}$$

برای به دست آوردن زبان تولید شده برای گرامر G، باید نحوه به دست آمدن رشته x از گرامر G مشخص شود. برای این منظور از مفهوم اشتقاق [۹] استفاده می‌شود. نخستین گام در یک اشتقاق، استفاده از یکی از قواعد متناظر با

[5]Chomsky [6]Right Linear Grammar [7]Left Linear Grammar [8]Regular Grammar [9]Derivation

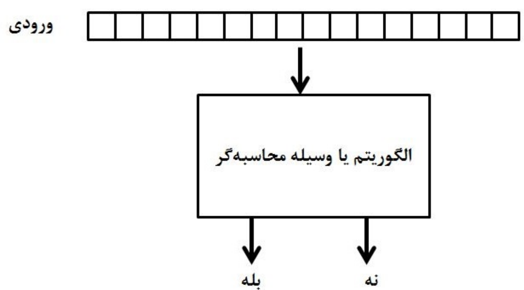
متغیر شروع است. گام‌های بعدی با جایگزینی متغیرهای ظاهرشده توسط یکی از قواعد متناظر با آن‌ها، تا رسیدن به رشته x ادامه می‌یابد. به عنوان مثال $S \Rightarrow Sb \Rightarrow Sbb \Rightarrow bb$ یک اشتقاق برای رشته bb از گرامر $G : S \to \lambda \mid Sb$ است. وجود اشتقاق برای رشته $x \in \Sigma^*$ در گرامر G با نماد $S \Rightarrow^* x$ نشان داده می‌شود. زبان تولید شده برای گرامر G عبارت است از:

$$L(G) = \{x \in \Sigma^* : \quad S \Rightarrow^* x\}$$

تعریف ۴.۲. *فرض کنید Σ یك الفبای دلخواه باشد. زبان $L \subseteq \Sigma^*$، یك زبان نوع ۳ از دسته‌بندی چامسکی است هرگاه گرامر منظم G موجود باشد که $L = L(G)$.*

توصیف مکانیکی (انتزاعی)، زبان‌های نوع ۳ از دسته بندی چامسکی توسط اتوماتون متناهی انجام می‌شود. در ادامه مفهوم اتوماتون متناهی قطعی و غیرقطعی به صورت خلاصه یادآوری می‌شود [۳، ۶].

اگر Σ یک الفبا باشد. یک اتوماتون متناهی قطعی، به صورت شهودی وسیله محاسبه‌گری (الگوریتمی) است که به ازای هر رشته $x \in \Sigma^*$، به عنوان ورودی، خروجی «بله» یا «نه» را تولید می‌کند، و زبان پذیرفته‌شده برای یک اتوماتون متناهی قطعی، مجموعه تمام رشته‌هایی است که به ازای آن‌ها خروجی بله تولید می‌شود. اتوماتون متناهی قطعی با استفاده از نمادهای ریاضی (بخصوص توابع) تعریف می‌شود (شکل ۱ را ببینید).



شکل ۱: اتوماتون متناهی

تعریف ۵.۲. *یک اتوماتون متناهی قطعی*[10] *پنج تایی $M = (Q, \Sigma, q_\circ, A, \delta)$ است که در آن:*

- *Q یک مجموعه متناهی است که آن‌را مجموعه حالت‌های M می‌نامند.*
- *Σ یک الفبا است.*
- *$q_\circ \in Q$ حالت شروع است. محاسبه در حالت $q_\circ$ آغاز می‌شود.*
- *A یک زیرمجموعه از Q است که آن‌را مجموعه حالت‌های پایانی می‌نامند. در حالتی که محاسبه در حالت پایانی تمام شود به منزله رسیدن به جواب «بله» است. در غیر این‌صورت به منزله رسیدن به جواب «نه» است.*
- *δ یک تابع از $Q \times \Sigma$ به Q است که آن‌را تابع انتقال*[11] *می‌نامند. تابع δ نحوه عملکرد M را توصیف می‌کند.*

هر اتوماتون متناهی $M = (Q, \Sigma, q_\circ, A, \delta)$ به ازای هر رشته ورودی $x \in \Sigma^*$، در حالت شروع $q_\circ$ با خواندن اولین نماد x محاسبه خود را شروع می‌کند. برحسب تابع انتقال خود، تغییر حالت می‌دهد و نمادهای بعدی را یکی پس از

[10]Deterministic Finite Automaton [11]Transition Function

http://dx.doi.org/10.22108/msci.2021.125480.1393

دیگری می‌خواند. در نهایت به انتهای رشته می‌رسد که محاسبه را در این مرحله با حالت پایانی یا حالت غیرپایانی برای رشته x تمام می‌کند. برای به دست آوردن توصیف دقیق مفهوم شهودی محاسبه انجام شده توسط اتوماتون متناهی قطعی توسیع تابع انتقال $\delta : Q \times \Sigma \to Q$ به تابع $\delta^* : Q \times \Sigma^* \to Q$ لازم است که با استفاده از استقرا انجام می‌شود.

تعریف ۶.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی قطعی باشد. دراین‌صورت تابع انتقال توسعه یافته[۱۲] $\delta^* : Q \times \Sigma^* \to Q$ به صورت استقرایی زیر تعریف می‌شود:*

- *به ازای هر $q \in Q$، $\delta^*(q, \lambda) = q$.*
- *$\delta^*(q, y\sigma) = \delta(\delta^*(q, y), \sigma)$ به ازای هر $q \in Q$، $y \in \Sigma^*$ و $\sigma \in \Sigma$.*

با استفاده از تابع انتقال توسعه یافته، تعریف ریاضی زبان پذیرفته‌شده برای اتوماتون متناهی قطعی بیان می‌شود.

تعریف ۷.۲. *الف) فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی قطعی باشد. زبان پذیرفته‌شده برای M که با نماد $L(M)$ نشان داده می‌شود، عبارت است از:*

$$L(M) = \{x \in \Sigma^* : \quad \delta^*(q_\circ, x) \in A\}$$

ب) زبان $L \subseteq \Sigma^*$ محاسبه‌پذیر توسط یک اتوماتون متناهی قطعی است هرگاه اتوماتون متناهی قطعی مانند $M = (Q, \Sigma, q_\circ, A, \delta)$ موجود باشد که $L = L(M)$.

در هر اتوماتون متناهی قطعی، به ازای هر $p \in Q$ و $\sigma \in \Sigma$، $\delta(p, \sigma)$ تعریف شده است. پس در طول محاسبات مربوط به هر رشته $x \in \Sigma^*$ هیچ انتخابی وجود ندارد؛ لذا هر محاسبه‌ای با پردازش کامل رشته x در یک حالت به پایان می‌رسد. با کنار گذاشتن این محدودیت در تابع انتقال، وجود انتخاب در محاسبات اتوماتون متناهی غیرقطعی ممکن می‌شود.

تعریف ۸.۲. *یک اتوماتون متناهی غیرقطعی[۱۳] پنج تایی $M = (Q, \Sigma, q_\circ, A, \delta)$ است که در آن:*

- *Q یک مجموعه متناهی است که آن‌را مجموعه حالت‌های M می‌نامند.*
- *Σ یک الفبا است.*
- *$q_\circ \in Q$ حالت شروع است.*
- *A یک زیرمجموعه از Q است که آن‌را مجموعه حالت‌های پایانی می‌نامند.*
- *$\delta : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ تابع انتقال M است که در آن $\mathcal{P}(Q)$ مجموعه تمام زیرمجموعه‌های Q است.*

تبصره ۹.۲. *توضیح نکات زیر در مورد تابع انتقال اتوماتون متناهی غیرقطعی $M = (Q, \Sigma, q_\circ, A, \delta)$ ضروری است:*

- *اگر $\delta(q, \sigma) = \emptyset$، آنگاه M در حالت q با خواندن نماد σ، هیچ تبدیل حالت نخواهد داشت و محاسبه متوقف می‌شود.*
- *اگر $\delta(q, \sigma) = \{p_۱, \ldots, p_n\}$، آنگاه M در حالت q با خواندن نماد σ، امکان تبدیل حالت به حالت‌های $p_۱$، ... و p_n (به عبارت دیگر n انتخاب) را دارد.*
- *چون λ رشته پوچ است، لذا $\delta(q, \lambda) = \{p_۱ \ldots, p_n\}$ به معنای این است که M در حالت q امکان تغییر حالت به حالت‌های $p_۱$، ... و p_n را بدون خواندن نمادی از رشته ورودی دارد. در اصطلاح M در حالت q، روی رشته ورودی در جا می‌زند؛ ولی حالت آن می‌تواند به یکی از حالت‌های $p_۱$، ... و p_n تغییر کند.*

---
[12] Extended Transition Function [13] Nondeterministic Finite Automaton

فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. در این‌صورت منظور از زبان پذیرفته شده برای M مجموعه xهای متعلق به Σ^* است که M در حالت شروع با خواندن نمادهای x از سمت چپ یکی پس از دیگری، بر حسب تابع انتقال δ، حداقل از یکی از مسیرهای محاسبه به حالت پایانی برسد. برای توصیف ریاضی زبان پذیرفته شده برای M، $L(M)$، توسیع تابع انتقال $\delta : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ به تابع $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ لازم است. با توجه به اینکه این امکان وجود دارد که برای $n \in \mathbb{N}$، $\delta^n(p, \lambda)$ یک مجموعه‌ای با بیش از یک عضو باشد، بنابراین برای تعریف تابع δ^* تعریف زیر مورد نیاز است.

تعریف ۱۰.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. در این‌صورت برای هر $q \in Q$، $\delta^n(q, \lambda)$ با استفاده از استقرا روی n به صورت زیر تعریف می‌شود:*

$$\delta^1(q, \lambda) = \delta(q, \lambda) \cup \{q\}$$
$$\delta^{n+1}(q, \lambda) = \delta^n(q, \lambda) \cup \{\delta^1(p, \lambda) : \quad p \in \delta^n(q, \lambda)\}$$

تعریف ۱۱.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. در این‌صورت تابع انتقال توسعه یافته $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ به صورت استقرایی زیر تعریف می‌شود:*

- *به ازای هر $q \in Q$، قرار دهید :*

$$\delta^*(q, \lambda) = \delta^n(q, \lambda)$$

که در آن $\delta^n(q, \lambda) = \delta^{n+1}(q, \lambda)$.

- *به ازای هر $q \in Q$، هر $y \in \Sigma^*$ و هر $\sigma \in \Sigma$ قرار دهید*

$$\delta^*(q, y\sigma) = \bigcup_{p \in \delta^*(q,y)} \delta(p, \sigma) \cup \bigcup_{p \in \delta^*(q,y)} \delta^*(\delta(p, \sigma), \lambda)$$

تعریف ۱۲.۲. *الف) فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. زبان پذیرفته شده برای M که با نماد $L(M)$ نشان داده می‌شود، عبارت است از:*

$$L(M) = \{x \in \Sigma^* : \quad \delta^*(q_\circ, x) \cap A \neq \emptyset\}$$

ب) زبان $L \subseteq \Sigma^*$، محاسبه‌پذیر توسط اتوماتون متناهی غیرقطعی گفته می‌شود هرگاه اتوماتون متناهی غیرقطعی مانند $M = (Q, \Sigma, q_\circ, A, \delta)$ موجود باشد که $L = L(M)$.

می‌توان نشان داد که مجموعه زبان‌های محاسبه‌پذیر توسط اتوماتای متناهی غیرقطعی، همان مجموعه زبان‌های محاسبه‌پذیر توسط اتوماتای متناهی قطعی است. به عبارت دیگر اگر چه در اتوماتای متناهی غیرقطعی مسیرهای محاسباتی مختلف وجود دارد؛ ولی نسبت به اتوماتای متناهی قطعی قدرت محاسباتی بیشتری را ندارد و تنها انجام محاسبات را ساده‌تر می‌کند.

قضیه ۱۳.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی قطعی باشد. دراین‌صورت اتوماتون متناهی غیرقطعی $M_1 = (Q, \Sigma, q_\circ, A, \delta_1)$ موجود است که $L(M) = L(M_1)$.*

اثبات. فرض کنید $\delta_1 : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ تابعی با ضابطه $\delta_1(q, \sigma) = \{\delta(q, \sigma)\}$ و $\delta_1(q, \lambda) = \emptyset$ باشد. واضح است که به ازای هر $q \in Q$ ، $\delta_1^*(q, \lambda) = \{q\}$ و به ازای هر $x \in \Sigma^*$، $\{\delta^*(q, x)\}$ یک مجموعه تک عضوی است. بنابراین با استقرا روی پیچیدگی رشته $x \in \Sigma^*$ می‌توان نشان داد که به ازای هر $q \in Q$، $\delta_1^*(q, x) = \{\delta^*(q, x)\}$. پس $L(M) = L(M_1)$. □

http://dx.doi.org/10.22108/msci.2021.125480.1393

در ادامه نشان داده می‌شود که عکس قضیه ۱۳.۲ نیز برقرار است. در واقع ابتدا هر اتوماتون متناهی غیرقطعی، تبدیل به اتوماتون متناهی غیرقطعی می‌شود که به ازای هر $p \in Q$، $\delta(p, \lambda) = \emptyset$. در اصطلاح هر اتوماتون متناهی غیرقطعی، تبدیل به اتوماتون متناهی غیرقطعی بدون λ–انتقال می‌شود. سپس اتوماتون حاصل به یک اتوماتون متناهی قطعی تبدیل می‌شود.

قضیه ۱۴.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. در این‌صورت یک اتوماتون متناهی غیرقطعی بدون λ–انتقال مانند $M_1 = (Q, \Sigma, q_\circ, A_1, \delta_1)$ موجود است که $L(M) = L(M_1)$.*

اثبات. تابع $\delta_1 : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ تابع انتقالی است که به ازای هر $\sigma \in \Sigma$ و $q \in Q$، $\delta_1(q, \lambda) = \emptyset$ و $\delta_1(q, \sigma) = \delta^*(q, \sigma)$. همچنین قرار می‌دهیم:

$$A_1 := \begin{cases} A \cup \{q_\circ\} & \lambda \in L(M) \\ A & \lambda \notin L(M) \end{cases}$$

به وضوح M_1 یک اتوماتون متناهی غیرقطعی بدون λ–انتقال است. بنابراین $\delta_1^*(q_\circ, \lambda) = \{q_\circ\}$. لذا با توجه به تعریف A_1، $\lambda \in L(M)$ اگر و تنها اگر $\lambda \in L(M_1)$. با استقرا روی پیچیدگی رشته $x \in \Sigma^* \setminus \{\lambda\}$ می‌توان ثابت کرد که $\delta_1^*(q_\circ, x) = \delta^*(q_\circ, x)$. بنابراین $L(M) = L(M_1)$. □

قضیه ۱۵.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی غیرقطعی باشد. دراین‌صورت یک اتوماتون قطعی مانند $M_1 = (Q_1, \Sigma, q_1, A_1, \delta_1)$ موجود است که $L(M) = L(M_1)$*

اثبات. طبق قضیه ۱۴.۲، بدون کاستن از کلیت می‌توان فرض کرد که M یک اتوماتون متناهی غیرقطعی بدون λ–انتقال است. برای ساخت M_1 از روی M قرار می‌دهیم $Q_1 := \mathcal{P}(Q)$، $q_1 := \{q_\circ\}$، $A_1 := \{q \subseteq Q : \ q \cap A \neq \emptyset\}$ و $\delta_1 : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$ تابع انتقالی است که به ازای هر $q \in \mathcal{P}(Q)$ و $\sigma \in \Sigma$، $\delta_1(q, \sigma) = \bigcup_{p \in q} \delta(p, \sigma)$. به وضوح M_1 یک اتوماتون متناهی قطعی است و به ازای هر $x \in \Sigma^*$، $\delta_1^*(q_\circ, x) = \delta^*(q_\circ, x)$. بنابراین $L(M_1) = L(M)$. □

در ادامه این بخش ارتباط بین زبان‌های نوع ۳ از دسته‌بندی چامسکی و زبان‌های محاسبه‌پذیر توسط اتوماتای متناهی بررسی و مطالعه می‌شود.

قضیه ۱۶.۲. *فرض کنید Σ یك الفبای دلخواه و $L \subseteq \Sigma^*$ توسط یك اتوماتون متناهی قطعی پذیرفته شود. در این‌صورت گرامر منظم مانند G موجود است که $L = L(G)$.*

اثبات. فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ اتوماتون متناهی قطعی باشد که $L = L(M)$. در این‌صورت گرامر منظم $G = (V, \Sigma, S, P)$ را در نظر بگیرید که در آن:

- $V = Q$.
- $S = q_\circ$.
- $P = \{q \to \sigma p : \ \ \delta(q, \sigma) = p\} \cup \{q \to \lambda : \ \ q \in A\}$.

به وضوح G یک گرامر خطی راست است. با استفاده از استقرا روی x می‌توان نشان داد که به ازای هر $p, q \in Q$، $\delta^*(q, x) = p$ اگر و تنها اگر اشتقاق $q \Rightarrow^* xp$ در G موجود باشد.

پس $x \in L$ اگر و تنها اگر یک اشتقاق به صورت $q_\circ \Rightarrow^* xq \Rightarrow x$ در G موجود باشد که $q \in A$. لذا $x \in L$ اگر و تنها اگر $x \in L(G)$. پس $L = L(G)$.

□

قضیه ۱۷.۲. *فرض کنید $G = (V, \Sigma, S, P)$ یک گرامر منظم باشد. دراین‌صورت $L(G)$ محاسبه‌پذیر توسط اتوماتون متناهی غیرقطعی است.*

اثبات. فرض کنید $G = (V, \Sigma, S, P)$ یک گرامر خطی راست باشد. در این‌صورت بدون کاستن از کلیت می‌توان فرض کرد که قواعد P به صورت $A \to \lambda$ و $A \to \sigma B$ است.

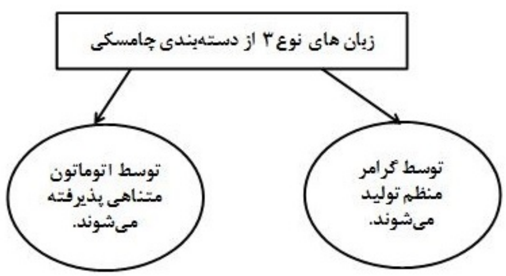
اتوماتون متناهی غیرقطعی $M = (Q, \Sigma, q_\circ, A, \delta)$ را درنظر بگیرید که در آن:

- $Q = V$.
- $q_\circ = S$.
- $A = \{q \in Q : \ q \to \lambda \in P\}$.
- $B \in \delta(A, \sigma)$ اگر و تنها اگر $A \to \sigma B$ در P موجود باشد.

با استقرا روی n می‌توان نشان داد که $S \Rightarrow^* \sigma_1 \cdots \sigma_n B_n$ اگر و تنها اگر $B_n \in \delta^*(S, \sigma_1 \cdots \sigma_n)$. بنابراین $L(G) = L(M)$.

به طور مشابه می‌توان نشان داد که زبان تولید شده برای گرامر خطی چپ نیز محاسبه‌پذیر توسط اتوماتون متناهی غیرقطعی است. □

طبق قضایای ۱۶.۲ و ۱۷.۲ مجموعه زبان‌های نوع ۳، دسته‌بندی چامسکی توسط گرامرهای منظم و اتوماتای متناهی تولید می‌شوند. به عبارت دیگر:



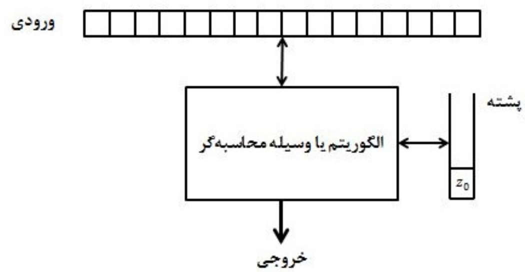
شکل ۲: زبان‌های نوع ۳ دسته‌بندی چامسکی

۳. زبان‌های نوع ۲ از دسته‌بندی چامسکی

در این بخش زبان‌های نوع ۲، دسته‌بندی چامسکی معرفی می‌شود [۲] ، [۱] و [۳] . تعریف مکانیکی آن‌ها با استفاده از اتوماتای پشته‌ای[14] انجام می‌شود. اتوماتای پشته‌ای از بسیاری جهات شبیه اتوماتای متناهی هستند؛ با این تفاوت که یک حافظه نامحدود در آن‌ها وجود دارد که بر اساس قوانین پشته[15] عمل می‌کند. مشابه اتوماتای متناهی تعریف این نوع ماشین‌ها با استفاده از توابع ریاضی انجام می‌شود (شکل ۳ را ببینید).

[14] Pushdown Automaton [15] Stack

http://dx.doi.org/10.22108/msci.2021.125480.1393



شکل ۳: اتوماتون پشته‌ای

تعریف ۱.۳. *یک اتوماتون پشته‌ای (غیر قطعی) هفت تایی $M = (Q, \Sigma, \Gamma, q_\circ, Z_\circ, A, \delta)$ است که در آن:*

- *Q یک مجموعه متناهی است که آن‌را مجموعه حالت‌های M می‌نامند.*
- *Σ الفبای مربوط به ورودی است. رشته‌های هر زبان مورد بحث در اینجا از Σ^* می‌باشند.*
- *Γ الفبای مربوط به پشته است. نمادهای مورد استفاده در پشته از Γ است.*
- *$q_\circ \in Q$ حالت شروع است. آغاز محاسبه در حالت $q_\circ$ انجام می‌شود.*
- *$Z_\circ \in \Gamma$ نماد شروع پشته است که خالی بودن پشته را نشان می‌دهد.*
- *A یک زیرمجموعه از Q است که آن‌را مجموعه حالات پایانی می‌نامند.*
- *$\delta : Q \times \Sigma \cup \{\lambda\} \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$ یک تابع است که آن‌را تابع انتقال می‌نامند. تابع δ نحوه عملکرد M را توصیف می‌کند.*

تبصره ۲.۳. *در هر اتوماتون متناهی (قطعی و غیرقطعی)، هر حرکت لحظه‌ای وابسته به حالت و نماد خوانده شده از ورودی است، در حالی که در اتوماتون پشته‌ای، هر حرکت لحظه‌ای وابسته به حالت، نماد خوانده شده از ورودی و آخرین نماد موجود در پشته است.*

- *$\delta(p, \sigma, X) = (q, Y)$ به معنای این است که اتوماتون پشته‌ای در حالت p با خواندن نماد σ از ورودی و خواندن X از پشته حالت خود را به q تبدیل می‌کند، روی رشته ورودی به سراغ نماد بعد از σ می‌رود و آخرین نماد X در پشته را برحسب Y تغییر می‌دهد. بیشترین تغییرات مورد استفاده در نمادهای پشته به صورت زیر است:*
  - *اگر $Y = \lambda$، آنگاه نماد X را از پشته پاک می‌کند و نماد قبل از آن آخرین نماد پشته می‌شود.*
  - *اگر $Y = X$، آنگاه هیچ تغییری در پشته ایجاد نمی‌شود.*
  - *اگر $Y = ZX$، آنگاه نماد Z به پشته اضافه می‌شود و آخرین نماد پشته می‌شود.*
  - *اگر $Y = Z$، آنگاه نماد X از پشته پاک می‌شود و به جای آن نماد Z به پشته اضافه می‌شود.*
- *$\delta(p, \lambda, X) = (q, Y)$ به معنای این است که اتوماتون پشته‌ای در حالت p بدون خواندن نمادی از رشته ورودی و خواندن X از پشته، حالت خود را به q تبدیل کرده و محتوای پشته را بر حسب Y تغییر می‌دهد.*

دنبال کردن حرکات یک اتوماتون پشته‌ای، روی یک رشته ورودی، پیچیده‌تر از دنبال کردن حرکات یک اتوماتون متناهی است. در اتوماتون متناهی، دنباله حرکات با استفاده از توسیع تابع انتقال δ به تابع δ^* به صورت دقیق بیان می‌شود؛ ولی در مورد اتوماتون پشته‌ای توسیع تابع انتقال دشوار و پیچیده است. بنابراین در اینجا از مفهوم پیکربندی استفاده می‌شود.

http://dx.doi.org/10.22108/msci.2021.125480.1393

فرض کنید $M = (Q, \Sigma, \Gamma, q_\circ, Z_\circ, A, \delta)$ یك اتوماتون پشته‌ای باشد. سه تایی (q, x, α) یک پیکربندی از M نامیده می‌شود که در آن $q \in Q$، $x = \sigma_1 \cdots \sigma_m \in \Sigma^*$ و $\alpha = \gamma_1 \cdots \gamma_n Z_\circ \in \Gamma^*$. می‌توان گفت هر پیکربندی توصیف لحظه‌ای از M است. یک گام از حرکت M با نماد $(p, x, \xi\alpha) \vdash_M (q, y, \gamma\beta)$ نشان داده می‌شود که به معنای این است که با استفاده از یکی از حرکت‌های ممکن M، پیکربندی اول به پیکربندی دوم تبدیل شده است. به طور دقیق‌تر، اگر $x = y$، آنگاه از حرکت $(q, \gamma) \in \delta(p, \lambda, \xi)$ در تغییر پیکربندی استفاده شده است. اگر $x = \sigma y$، آنگاه از حرکت $(q, \gamma) \in \delta(p, \sigma, \xi)$ در تغییر پیکربندی استفاده شده است. در حالت کلی نماد $(p, x, \alpha) \vdash_M^* (q, y, \beta)$ به معنای این است $n \in \mathbb{N}$ موجود است که با دنباله‌ای از n حرکت، پیکربندی اول به پیکربندی دوم تبدیل شده است.

پذیرش رشته x، توسط اتوماتون پشته‌ای M زمانی اتفاق می‌افتد که حداقل یک دنباله از حرکات در M موجود باشد که طی آن حرکات تمام نمادهای موجود در x، خوانده شده و M در حالت پایانی متوقف شده باشد.

تعریف ۳.۳. *فرض کنید $M = (Q, \Sigma, \Gamma, q_\circ, Z_\circ, A, \delta)$ یک اتوماتون پشته‌ای باشد. زبان پذیرفته شده برای M که با نماد $L(M)$ نشان داده می‌شود، عبارت است از:*

$$L(M) = \bigcup_{q \in A} \{x \in \Sigma^* : \quad (q_\circ, x, Z_\circ) \vdash_M^* (q, \lambda, \alpha); \quad \alpha \in \Gamma^*\}$$

هر اتوماتون متناهی را می‌توان یك اتوماتون پشته‌ای در نظر گرفت که در آن از پشته استفاده نمی‌شود.

گزاره ۴.۳. *هر زبان پذیرفته شده برای اتوماتون متناهی، توسط یك اتوماتون پشته‌ای پذیرفته می‌شود.*

اثبات. فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک اتوماتون متناهی قطعی باشد. در این‌صورت اتوماتون پشته‌ای مانند $M_1 = (Q, \Sigma, \Gamma = \{Z_\circ\}, q_\circ, Z_\circ, A, \delta_1)$ موجود است که در آن $\delta_1(q, \sigma, Z_\circ) = (\delta(q, \sigma), Z_\circ)$. به وضوح $L(M) = L(M_1)$. □

تبصره ۵.۳. *زبان $L = \{a^n b^n : \ n \in \mathbb{N}\}$ توسط اتوماتای پشته‌ای پذیرفته می‌شود؛ در حالی‌که توسط هیچ اتوماتون متناهی پذیرفته نمی‌شود [۳]. بنابراین طبق گزاره ۴.۳، مجموعه زبان‌های پذیرفته‌شده برای اتوماتای متناهی یک زیرمجموعه محض از مجموعه زبان‌های پذیرفته‌شده برای اتوماتای پشته‌ای است (شکل ۴ را ببینید).*

شکل ۴: زبان‌های پذیرفته شده برای اتوماتون پشته‌ای و اتوماتون متناهی

گرامرهای مستقل از متن[16] برای توصیف صوری زبان‌های پذیرفته شده برای اتوماتای پشته‌ای غیرقطعی استفاده می‌شوند.

[16] Context Free Grammar

http://dx.doi.org/10.22108/msci.2021.125480.1393

یک گرامر مستقل از متن، مانند گرامر منظم چهارتایی $G=(V,\Sigma,S,P)$ است با این تفاوت که هر قاعده در آن به صورت $A\to\alpha$ است که $A\in V$ و $\alpha\in(\Sigma\cup V)^*$. زبان تولید شده برای G را که با نماد $L(G)$ نشان داده می‌شود، عبارت است از $L(G)=\{x\in\Sigma^*:\quad S\Rightarrow^* x\}$. همچنین زبان $L\subseteq\Sigma^*$ را مستقل از متن[17] گویند هرگاه گرامر مستقل از متن G موجود باشد که $L=L(G)$. هر زبان مستقل از متن یک زبان نوع ۲ از دسته‌بندی چامسکی است.

تعریف ۶.۳. *فرض کنید $G=(V,\Sigma,S,P)$ یک گرامر مستقل از متن باشد.* $M_G=(Q,\Sigma,\Gamma,q_0,Z_0,A,\delta)$ یک *اتوماتون پشته‌ای متناظر با G است که در آن $Q=\{q_0,q_1,q_2\}$، $\Gamma=V\cup\Sigma\cup\{Z_0\}$، $A=\{q_2\}$ و تابع انتقال $\delta:Q\times\Sigma\cup\{\lambda\}\times\Gamma\to\mathcal{P}(Q\times\Gamma^*)$ به صورت زیر تعریف می‌شود:*

$$\delta(q_0,\lambda,Z_0)=\{(q_1,SZ_0)\}$$
$$\delta(q_1,\lambda,Z_0)=\{(q_2,Z_0)\}$$

به ازای هر $A\in V$ و $\sigma\in\Sigma$،

$$\delta(q_1,\lambda,A)=\{(q_1,\alpha):\ \ \exists A\to\alpha\in P\}$$
$$\delta(q_1,\sigma,\sigma)=\{(q_1,\lambda)\}$$

قضیه ۷.۳. *اگر $G=(V,\Sigma,S,P)$ یک گرامر مستقل از متن و M_G اتوماتون پشته‌ای متناظر با آن باشد، آنگاه $L(G)=L(M_G)$.*

اثبات. فرض کنید $x=x_0x_1\cdots x_{m+1}\in L(G)$ که $x_0,\ldots,x_{m+1}\in\Sigma^*$. در این‌صورت یک اشتقاق برای x در G به صورت زیر موجود است:

$$S\Rightarrow x_0A_0\alpha_0\Rightarrow x_0x_1A_1\alpha_1\Rightarrow\cdots\Rightarrow x_0x_1\cdots x_mA_m\alpha_m\Rightarrow x_0x_1\cdots x_{m+1}$$

که در آن به ازای هر $0\le i\le m$، $A_i\in V$ و $\alpha_i\in(\Sigma\cup V)^*$.

با دنباله‌ای از حرکات M_G، اشتقاق فوق، به صورت زیر شبیه سازی می‌شود:

مرحله ۱ (شروع استقرا). اولین گام اشتقاق فوق (یعنی $S\Rightarrow x_0A_0\alpha_0$) با مجموعه حرکات زیر شبیه سازی می‌شود که حاصل آن خواندن پیشوند x_0 از رشته ورودی x و ذخیره $A_0\alpha_0$ در پشته است.

$$(q_0,x_0x_1\cdots x_{m+1},Z_0)\vdash_{M_G}(q_1,x_0x_1\cdots x_{m+1},SZ_0)\vdash_{M_G}(q_1,x_0x_1\cdots x_{m+1},x_0A_0\alpha_0Z_0)\vdash_{M_G}(q_1,x_1\cdots x_{m+1},A_0\alpha_0Z_0)$$

مرحله ۲ (فرض استقرا). فرض کنید برای هر $i<m$، با یک مجموعه حرکات در M_G ، iگام از اشتقاق را شبیه سازی می‌شود. یعنی دنباله‌ای از حرکات در M_G موجود است که حاصل آن خواندن پیشوند $x_0\cdots x_i$ از رشته ورودی x و ذخیره سازی $A_i\alpha_i$ در پشته است. به عبارت دیگر $(q_0,x_0x_1\cdots x_{m+1},Z_0)\vdash^*_{M_G}(q_1,x_{i+1}\cdots x_{m+1},A_i\alpha_iZ_0)$.

مرحله ۳(حکم استقرا). حرکات چنان شبیه سازی می‌شود که یک گام بیشتر از اشتقاق فوق را که منجر به خواندن پیشوند $x_0\cdots x_{i+1}$ از رشته ورودی x و ذخیره سازی $A_{i+1}\alpha_{i+1}$ در پشته شود. به عبارت دیگر:

$$(q_0,x_0x_1\cdots x_{m+1},Z_0)\vdash^*_{M_G}(q_1,x_{i+1}\cdots x_{m+1},A_i\alpha_iZ_0)\vdash_{M_G}(q_1,x_{i+1}\cdots x_{m+1},x_{i+1}A_{i+1}\alpha_{i+1}Z_0)\vdash_{M_G}(q_1,x_{i+2}\cdots x_{m+1},A_{i+1}\alpha_{i+1}Z_0)$$

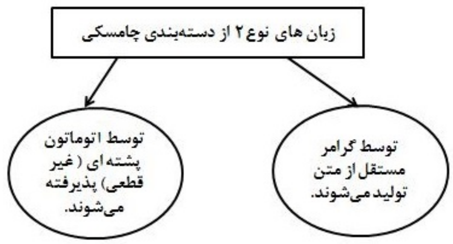
بنابراین به ازای $i=m-1$، طبق فرض استقرا و آخرین گام اشتقاق مجموعه حرکات زیر وجود دارد:

[17] Context Free Language

$$(q_0, x_0x_1\cdots x_{m+1}, Z_0) \vdash^*_{M_G} (q_1, x_mx_{m+1}, A_{m-1}\alpha_{m-1}Z_0) \vdash_{M_G} (q_1, x_mx_{m+1}, x_mA_m\alpha_mZ_0) \vdash_{M_G}$$
$$(q_1, x_{m+1}, A_m\alpha_mZ_0) \vdash_{M_G} (q_1, x_{m+1}, x_{m+1}Z_0) \vdash_{M_G} (q_1, \lambda, Z_0) \vdash_{M_G} (q_2, \lambda, Z_0)$$

پس $(q_0, x, Z_0) \vdash^*_{M_G} (q_2, \lambda, Z_0)$. لذا $x \in L(M_G)$.
برعکس به ازای هر $x \in L(M_G)$، به طور مشابه هر دنباله از حرکات M_G که منجر به پذیرش آن می‌شود، با یک اشتقاق برای x در گرامر G شبیه سازی می‌شود. بنابراین $x \in L(G)$. □

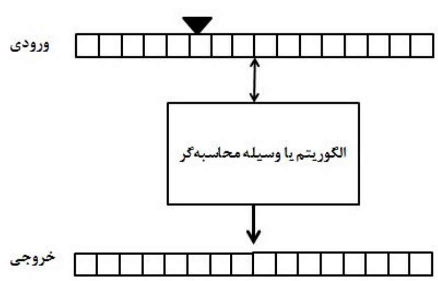
لازم به توضیح است که برای هر اتوماتون پشته‌ای مانند M می‌توان گرامر مستقل از متن G را به دست آورد که $L(G) = L(M)$ [۴، بخش ۵٫۴]. لذا طبق قضیه ۷.۳ زبان‌های از نوع ۲، دسته‌بندی چامسکی توسط گرامرهای مستقل از متن و اتوماتای پشته‌ای غیرقطعی تولید می‌شوند. به عبارت دیگر:



شکل ۵: زبان‌های نوع ۲ دسته‌بندی چامسکی

۴. زبان‌های نوع ۰ از دسته‌بندی چامسکی

در این بخش زبان‌های نوع ۰ از دسته‌بندی چامسکی معرفی می‌شود. این نوع زبان‌ها توسط ماشین‌های تورینگ[18] (توسیعی از اتوماتای پشته‌ای) پذیرفته می‌شوند و گرامرهای نامقید آن‌ها را تولید می‌کنند [۴] و [۳].
یک ماشین تورینگ یک اتوماتون پشته‌ای است که حافظه آن یک نوار تقسیم‌بندی شده به سلول‌ها است، به‌طوری‌که هر سلول برای نگهداری یک نماد به کار می‌رود. در ارتباط با نوار یک هد[19] موجود است که می‌تواند به راست یا چپ حرکت کند (شکل ۶ را ببینید).



شکل ۶: ماشین تورینگ

[18] Turing Machine [19] Head Tape

http://dx.doi.org/10.22108/msci.2021.125480.1393

تعریف ۱.۴. *یک ماشین تورینگ پنج‌تایی $T = (Q \cup \{h\}, \Sigma, \Gamma \cup \{B\}, q_\circ, \delta)$ است که در آن:*

- *Q یک مجموعه متناهی است که آن‌را مجموعه حالت‌های T می‌نامند.*
- *h حالت توقف پایانی است. اگر T در مرحله‌ای از محاسبه خود به حالت h برسد به منزله توقف و پذیرش است.*
- *$q_\circ \in Q$ حالت شروع است. محاسبه در حالت $q_\circ$ آغاز می‌شود.*
- *Σ مجموعه الفبای ورودی است. رشته‌های زبان‌های مورد بحث اعضای Σ^* هستند.*
- *Γ یک مجموعه متناهی است که الفبای نوار است و $\Sigma \subseteq \Gamma$.*
- *نماد B خالی بودن سلول را در نوار نشان می‌دهد.*
- *δ تابعی از $Q \times \Gamma \cup \{B\}$ به مجموعه $\mathcal{P}((Q \cup \{h\}) \times \Gamma \cup \{B\} \times \{L, R, S\})$ است که عملکرد T را مشخص می‌کند.*

تبصره ۲.۴. *در مورد عملکرد تابع انتقال δ، $(q, Y, D) \in \delta(p, X)$ به معنای این است که اگر حالت T، p باشد و هد روی سلولی حاوی X قرار گیرد، آنگاه حالت p تبدیل به q می‌شود، محتوای سلول از X به Y تبدیل می‌شود و هد برحسب D، به راست یا چپ حرکت می‌کند یا بدون حرکت روی همان سلول باقی می‌ماند. همچنین اگر $q = h$، آنگاه T در حالت پایانی h متوقف می‌شود.*

تبصره ۳.۴. *نماد $\tau_1 \cdots \tau_n q \sigma_1 \cdots \sigma_m$ وضعیت لحظه‌ای ماشین تورینگ T را توصیف می‌کند که پیکربندی (ماشین تورینگ) گفته می‌شود، و نشانگر این است که محتوای نوار (به ترتیب از چپ به راست) $\tau_1 \cdots \tau_n \sigma_1 \cdots \sigma_m$، حالت ماشین q و هد نیز روی سلولی از نوار حاوی نماد σ_1، واقع است. دنباله حرکات ماشین تورینگ T با استفاده از پیکربندی‌ها توصیف می‌شوند. نماد $\tau_1 \cdots \tau_n q \sigma_1 \cdots \sigma_m \vdash^* \tau'_1 \cdots \tau'_s q' \sigma'_1 \cdots \sigma'_t$ نشانگر تبدیل پیکربندی اول به پیکربندی دوم در تعداد متناهی گام است. لازم به توضیح است که اگر $x = \sigma_1 \cdots \sigma_n$ یک رشته ورودی باشد، پیکربندی آغازین برای محاسبات مربوط به x در T به صورت $q_\circ B \sigma_1 \cdots \sigma_n$ است.*

تعریف ۴.۴. *فرض کنید $T = (Q \cup \{h\}, \Sigma, \Gamma \cup \{B\}, q_\circ, \delta)$ یک ماشین تورینگ باشد و $x \in \Sigma^*$.*

- *ماشین T رشته x را **می‌پذیرد** هرگاه $q_\circ Bx \vdash^* yhw$ که $y, w \in (\Gamma \cup \{B\})^*$.*
- *ماشین T برای رشته x **متوقف می‌شود** هرگاه $q_\circ Bx \vdash^* yq\sigma w$ که $q \in Q$، $\sigma \in \Sigma \cup \{B\}$، $y, w \in (\Gamma \cup \{B\})^*$ و $\delta(q, \sigma)$ تعریف نشده باشد.*

برخلاف اتوماتون متناهی و پشته‌ای، پردازش کامل رشته ورودی $x \in \Sigma^*$، برای پذیرش آن در ماشین‌های تورینگ ضروری نیست. یک ماشین تورینگ در طول محاسبات مربوط به رشته ورودی یا در حالت پایانی یا در حالت غیرپایانی متوقف می‌شود. همچنین این امکان وجود دارد که هرگز متوقف نشود، انجام یک حلقه از محاسبات تکراری در مرحله‌ای از محاسبه باعث ایجاد چنین شرایطی می‌شود.

تعریف ۵.۴. *فرض کنید $T = (Q \cup \{h\}, \Sigma, \Gamma \cup \{B\}, q_\circ, \delta)$ یک ماشین تورینگ باشد. زبان پذیرفته‌شده برای T که با نماد $L(T)$ نشان داده می‌شود، عبارت است از:*

$$L(T) = \{x \in \Sigma^* : \quad q_\circ Bx \vdash^* yhw\}$$

تعریف ۶.۴. *زبان $L \subseteq \Sigma^*$ را به‌طور بازگشتی شمارا [20] گفته می‌شود هرگاه ماشین تورینگی مانند T موجود باشد که $L = L(T)$.*

[20]Enumerable Recursively

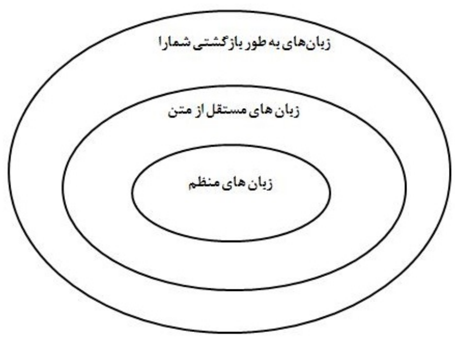
هر زبان به‌طور بازگشتی شمارا، یك زبان نوع ۰ از دسته‌بندی چامسکی است. توصیف صوری زبان‌های به‌طور بازگشتی شمارا، توسط گرامرهای نامقید انجام می‌شود. یک گرامر نامقید، مانند گرامر مستقل از متن چهارتایی $G = (V, \Sigma, S, P)$ است که هر قاعده در آن به صورت $\alpha \to \beta$ است که $\alpha, \beta \in (\Sigma \cup V)^*$ و رشته α شامل حداقل یک متغیر است. زبان تولید شده برای گرامر نامقید G عبارت است از:

$$L(G) = \{x \in \Sigma^* : \quad S \Rightarrow^* x\}$$

به وضوح هر گرامر مستقل از متن یک گرامر نامقید است؛ در حالی‌که گرامرهای نامقیدی موجودند که مستقل از متن نیستند. به عنوان مثال گرامر زیر تولید کننده زبان $L = \{a^{2^n} : \quad n \in \mathbb{N}\}$ است که مستقل از متن نیست [۴].

$$G : \begin{cases} S \to LaR \\ L \to LD \mid \lambda \\ Da \to aaD \\ DR \to R \\ R \to \lambda \end{cases}$$

بنابراین ارتباط زبان‌های مورد بحث به صورت شکل ۷ می‌باشد.



شکل ۷: ارتباط زبان‌های موردبحث

قضیه ۷.۴. *برای هر گرامر نامقید G، ماشین تورینگ T موجود است که $L(T) = L(G)$.*

اثبات. در ادامه عملکرد ماشین تورینگ T که پذیرنده زبان $L(G)$ است، به طور مختصر توصیف می‌شود:

ــ الفبای ورودی شامل تمام متغیرها و ترمینال‌های موجود در G است، البته این امکان هم وجود دارد که در طول محاسبه بر حسب نیاز، نمادهای جدیدی نیز اضافه شوند.

ــ در گام اول، هد به سمت راست حرکت می‌کند تا به اولین سلول خالی، بعد از x برسد.

ــ در گام دوم، اشتقاق‌های گرامر G به طور غیرقطعی شبیه‌سازی می‌شود. در این مرحله در اولین سلول خالی بعد از x، S جایگزین می‌شود و در طول محاسبات رشته x تغییر نمی‌کند. دنباله حرکات زیر را در جهت شبیه سازی اشتقاق‌ها تکرار می‌شود:

۱ ــ قاعده $\alpha \to \beta$ از G انتخاب می‌شود.

http://dx.doi.org/10.22108/msci.2021.125480.1393

۲ــ در صورت وجود α در رشته بعد از x یک α انتخاب می‌شود.
۳ــ α انتخاب شده با β جایگزین می‌شود.
ــ در گام سوم رشته ایجاد شده در گام دوم با x مقایسه می‌شود. اگر با x یکی باشد، آنگاه T در حالت پایانی متوقف می‌شود، در غیر اینصورت رشته ایجاد شده پاک می‌شود و گام دوم و سوم با انتخاب‌های جدید تکرار می‌شود.
بنابراین واضح است که $L(T) = L(G)$. □

قضیه ۸.۴. *برای هر ماشین تورینگ T، گرامر نامقید G موجود است که $L(G) = L(T)$.*

اثبات. برای سادگی فرض کنید $\Sigma = \{a, b\}$. گرامر مورد نظر سه نوع قاعده مختلف دارد که به طور مختصر در ادامه توصیف می‌شود:
ــ دسته اول: این گروه از قواعد برای هر رشته $x = \sigma_1 \cdots \sigma_n \in \Sigma^*$، رشته $(\sigma_1\sigma_1)\cdots(\sigma_n\sigma_n)$ را تولید می‌کنند که شامل دو کپی از x به نام‌های x_1 و x_2 است که x_1 در طول محاسبه بدون تغییر می‌ماند.
قواعد دسته اول به صورت زیر هستند:

$$\begin{cases} S \to S(BB) \mid T \\ T \to T(aa) \mid T(bb) \mid q_0(BB) \end{cases}$$

به عنوان مثال برای رشته ورودی ba، پیکربندی آغازین q_0BbaB ماشین تورینگ T به صورت $q_0(BB)(bb)(aa)(BB)$ توسط قواعد دسته اول شبیه‌سازی می‌شود.
دسته دوم: مجموعه قواعدی خواهند بود که مجموعه حرکات T را روی x_2 شبیه‌سازی می‌کنند، و x_1 بدون تغییر باقی می‌ماند.
دسته سوم: مجموعه قواعدی هستند که برای $x \in L(T)$، امکان پاک کردن هر نمادی روی رشته نهایی به جز x_1 را فراهم می‌سازد.
دسته دوم و سوم از قواعد با استفاده از یک مثال توصیف می‌شود. فرض کنید جدول انتقال ماشین تورینگ T به صورت زیر باشد:

δ	B	a	b
q_0	(q_1, B, R)	$-$	$-$
q_1	$-$	(q_2, b, L)	(q_1, b, R)
q_2	(h, B, S)	$-$	(q_2, b, R)
h	$-$	$-$	$-$

با توجه به تابع انتقال قواعد دسته دوم حاصل می‌شود.
قاعده $q_0(BB) \to (BB)q_1$، انتقال $\delta(q_0, B) = (q_1, B, R)$ را شبیه‌سازی می‌کند. B دوم در هر دو جفت از B ها ضروری است. زیرا به عنوان نمادی از رشته ورودی فرض می‌شود . اما نماد B اول در هر دو جفت بدون تغییر می‌ماند. بنابراین آن‌ها می‌توانند هر نماد دیگری هم باشند. پس قواعد $q_0(aB) \to (aB)q_1$ و $q_0(bB) \to (bB)q_1$ نیز موجود هستند. پس با توجه به تابع انتقال مجموعه قواعد زیر برای دسته دوم قواعد حاصل می‌شوند:
برای هر $\sigma \in \{a, b, B\}$ داریم:

$$\delta(q_0, B) = (q_1, B, R) : q_0(\sigma B) \to (\sigma B)q_1$$

http://dx.doi.org/10.22108/msci.2021.125480.1393

برای هر $\sigma \in \{a,b,B\}$ داریم:

$$\delta(q_1,b)=(q_1,b,R) : q_1(\sigma b) \to (\sigma b)q_1$$

برای هر $\sigma \in \{a,b,B\}$ داریم:

$$\delta(q_2,b)=(q_2,b,R) : q_2(\sigma b) \to (\sigma b)q_2$$

برای هر $\sigma_1,\sigma_2,\sigma_3 \in \{a,b,B\}$ داریم:

$$\delta(q_1,a)=(q_2,b,L) : (\sigma_1\sigma_2)q_0(\sigma_3 a) \to q_2(\sigma_1\sigma_2)(\sigma_3 b)$$

برای هر $\sigma \in \{a,b,B\}$ داریم:

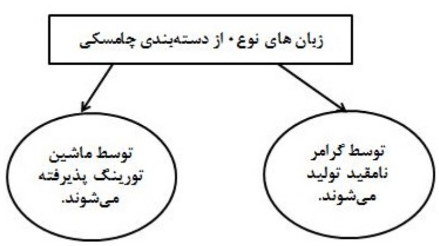
$$\delta(q_2,B)=(h,B,S) : q_2(\sigma B) \to h(\sigma B)$$

ظاهر شدن h در رشته حاصل از اشتقاق به معنای پذیرش رشته x توسط T است، در این مرحله با استفاده از قواعد دسته سوم تمام نمادها به جز نمادهای x_1 پاک می‌شوند. برای این منظور قواعد زیر لازم هستند:

$$\begin{cases} (\sigma_1\sigma_2)h \to h(\sigma_1\sigma_2)h \\ h(\sigma_1\sigma_2) \to h(\sigma_1\sigma_2)h \\ h(\sigma_1\sigma_2) \to \sigma_1 \\ h(B\sigma_2) \to \lambda \end{cases}$$

این مثال نوع قواعد مورد نیاز برای شبیه‌سازی حرکات هر ماشین تورینگ را به صورت اشتقاق مشخص می‌کند. در واقع با استفاده از قواعد نوع اول دو کپی x_1 و x_2 از یک رشته ورودی x حاصل می‌شود. با استفاده از قواعد دسته دوم عملیات ممکن T روی x_2 انجام می‌شود. در پایان با استفاده از قواعد دسته سوم در صورت پذیرش x توسط T، اشتقاق با حفظ نمادهای x_1 تمام می‌شود. $\square$

طبق قضایای ۷.۴ و ۸.۴ مجموعه زبان‌های نوع ۰، دسته‌بندی چامسکی توسط گرامرهای نامقید و ماشین‌های تورینگ تولید می‌شوند. به عبارت دیگر:



شکل ۸: زبان‌های نوع ۰ از دسته‌بندی چامسکی

http://dx.doi.org/10.22108/msci.2021.125480.1393

۵. زبان‌های نوع ۱ از دسته‌بندی چامسکی

زبان‌های از نوع ۱ دسته‌بندی چامسکی، توسط نوع خاصی از ماشین‌های تورینگ به نام اتوماتای کران‌دار خطی[۲۱] پذیرفته می‌شوند، و نوع خاصی از گرامرهای نامقید به نام گرامرهای حساس به متن[۲۲] تولید کننده آن‌ها هستند [۲، ۴].

تعریف ۱.۵. *یک اتوماتون کران‌دار خطی، ماشین تورینگ $T = (Q \cup \{h\}, \Sigma, \Gamma, q_0, \delta)$ است که در آن موارد زیر برقرار است:*

۱) *دو علامت مخصوص به‌نام‌های علامت انتهای چپ] و علامت انتهای راست [در Γ موجودند که اعضای Σ نیستند.*
۲) *در ابتدا پیکربندی آغازین متناظر با رشته x به $q_0[x]$ تبدیل می‌شود.*
۳) *در طول محاسبه حرکت هد به سمت چپ علامت انتهایی چپ] و حرکت آن به سمت راست نماد انتهایی راست [مجاز نیست.*
۴) *در طول محاسبه جایگزینی علامت‌های انتهایی ممکن نیست.*

تبصره ۲.۵. *در اتوماتای کران‌دار خطی مقدار نواری که استفاده می‌شود تابعی بر حسب طول رشته ورودی است. یعنی در اینجا روی فضای استفاده شده در پردازش یک رشته محدودیت وجود دارد.*

تعریف ۳.۵. *یک گرامر حساس به متن، گرامر نامقیدی است که هر قاعده در آن به فرم $\alpha \to \beta$ است که در آن $|\alpha| \leq |\beta|$. زبان تولید شده برای گرامر حساس به متن یک زبان حساس به متن نامیده می‌شود.*

قضیه ۴.۵. *اگر $L \subseteq \Sigma^*$ یک زبان حساس به متن باشد، آنگاه یك اتوماتون کران‌دار خطی موجود است که L را می‌پذیرد.*

اثبات. با ایجاد تغییرات جزئی در برهان قضیه ۷.۴ می‌توان اتوماتون کران‌دار خطی T را به دست آورد که $L(T) = L$. فرض کنید $G = (V, \Sigma, S, P)$ گرامر حساس به متن تولید کننده زبان L باشد. الفبای ورودی T شامل تمام متغیرها و ترمینال‌های موجود در G است، و در صورت نیاز نمادهای جدیدی اضافه می‌شوند. در گام اول پیکربندی $q_0[\sigma_1, \ldots, \sigma_n]$ به $q_0[(\sigma_1, B) \cdots (\sigma_n, B)]$ تبدیل می‌شود. در اینجا نیز دقیقاً روشی مشابه برهان قضیه ۷.۴ استفاده می‌شود؛ با این تفاوت که در قضیه ۷.۴ از فضای خالی موجود در نوار بعد از رشته ورودی x استفاده می‌شد؛ در حالی‌که در اینجا از مولفه‌های دوم به عنوان فضای کاربردی استفاده می‌شود. ابتدا نماد S به جای B در زوج مرتب (σ_1, B) جایگزین می‌شود. یعنی

$$q_0[(\sigma_1, B) \cdots (\sigma_n, B)] \vdash q_0[(\sigma_1, S) \cdots (\sigma_n, B)].$$

در ادامه مشابه گام‌های دوم و سوم از برهان قضیه ۷.۴ شبیه‌سازی اشتقاق‌های G را به صورت غیرقطعی انجام می‌دهد. پذیرش رشته x زمانی اتفاق می‌افتد که رشته ایجاد شده در مولفه‌های دوم با رشته موجود در مولفه اول یکسان باشد. هرگاه رشته ایجاد شده روی مولفه دوم با رشته موجود یکسان نباشد و یا اگر رشته‌ای با طول بیشتر از طول x ایجاد شود، آنگاه x پذیرفته نمی‌شود. □

قضیه ۵.۵. *اگر $T = (Q, \Sigma, \Gamma, q_0, \delta)$ یک اتوماتون کران‌دار خطی باشد، آنگاه گرامر حساس به متن G موجود است که $L(G) = L(T) \setminus \{\lambda\}$.*

[21] Linear Bounded Automaton [22] Context Sensitive Grammar

http://dx.doi.org/10.22108/msci.2021.125480.1393

اثبات. در ادامه ایده ساخت گرامر G از روی اتوماتون کران‌دار خطی T آورده می‌شود. با توجه به اینکه T یک ماشین تورینگ است. پس طبق قضیه ۸.۴، گرامر نامقید G_1 موجود است که $L(G_1) = L(T)$. با توجه به موارد زیر می‌توان G_1 را تبدیل به یک گرامر حساس به متن کرد:

۱ــ تمام قواعد ظاهر شده در G_1 به فرم قواعد یک گرامر حساس به متن هستند به جز قواعدی به فرم‌های $h(\sigma_1, \sigma_2) \to \sigma_1$ و $h(B\sigma_1) \to \lambda$. برای حل مشکل قواعدی به فرم $h(\sigma_1, \sigma_2) \to \sigma_1$ ، کافی است متغیرهای جدید $h(\sigma_1, \sigma_2)$ را به مجموعه متغیرهای G_1 اضافه می‌شود. در مورد قواعدی به فرم $h(B\sigma_1) \to \lambda$، چون هیچ پیکربندی از T با B شروع نمی‌شود. پس هر گز از این نوع قواعد استفاده نخواهد شد. بنابراین می‌توان آن‌ها را نادیده گرفت.

۲ــ به دلیل وجود علامت‌های انتهای چپ] و انتهای راست [در الفبای کاربردی G_1 ، متغیرهای جدید لازم است:

* برای هر $\sigma_1, \sigma_2 \in \Sigma \cup \{B\}$، متغیرهای $(\sigma_1[\sigma_2)$، $(\sigma_1\sigma_2])$ و $(\sigma_1[\sigma_2])$ را به مجموعه متغیرها اضافه می‌شود.

* متغیرهای به فرم $q(\sigma_1[\sigma_2)^L$، $q(\sigma_1[\sigma_2])^L$، $q(\sigma_1\sigma_2])^R$ و $q(\sigma_1[\sigma_2])^R$ را به مجموعه متغیرها اضافه می‌شود.

به عنوان مثال متغیر $q(\sigma_1[\sigma_2)^L$ نشانگر این است که q حالت T است، هد رو علامت انتهای چپ] قرار دارد و محتوای اولیه سلول σ_1 بوده که طی محاسبات به σ_2 تغییر کرده است.

۳ــ با توجه به اضافه شدن متغیرهای جدید، قواعد جدید مورد نیاز است که به دو مورد بر حسب تابع انتقال اشاره می‌شود:

* اگر $\delta(p, a) = (q, b, R)$، آنگاه قواعد متناظر با آن به صورت زیر خواهند بود:

$$
\begin{gathered}
p(\sigma_1 a)(\sigma_2\sigma_3) \to (\sigma_1 b)q(\sigma_2\sigma_3)\\
p(\sigma_1[a)(\sigma_2\sigma_3) \to (\sigma_1[b)q(\sigma_2\sigma_3)\\
p(\sigma_1 a)(\sigma_2\sigma_3]) \to (\sigma_1 b)q(\sigma_2\sigma_3])\\
p(\sigma_1[a)(\sigma_2\sigma_3]) \to (\sigma_1[b)q(\sigma_2\sigma_3])\\
p(\sigma_1 a]) \to q(\sigma_1 b])^R\\
p(\sigma_1[a]) \to q(\sigma_1[b])^R
\end{gathered}
$$

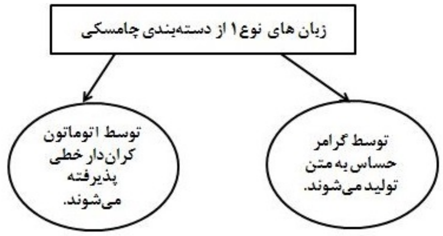
* اگر $\delta(p, [) = (q, [, R)$، آنگاه قواعد متناظر با آن به صورت زیر خواهند بود:

$$
\begin{gathered}
p(\sigma_1[\sigma_2)^L \to q(\sigma_1[\sigma_2)\\
p(\sigma_1[\sigma_2])^L \to q(\sigma_1[\sigma_2])\\
p(\sigma_1 a)(\sigma_2\sigma_3]) \to (\sigma_1 b)q(\sigma_2\sigma_3])\\
p(\sigma_1[a)(\sigma_2\sigma_3]) \to (\sigma_1[b)q(\sigma_2\sigma_3])\\
p(\sigma_1 a]) \to q(\sigma_1 b])^R\\
p(\sigma_1[a]) \to q(\sigma_1[b])^R
\end{gathered}
$$

بنابراین با توجه به توضیحات داده شده، مشابه برهان قضیه ۸.۴ می‌توان گرامر حساس به متن مورد نظر را به دست آورد. □

هر زبان تولیده شده توسط گرامر حساس به متن یك زبان از نوع ۰ دسته‌بندی چامسکی است. لذا طبق قضایای ۴.۵ و ۵.۵ این نوع زبان‌ها توسط اتوماتای خطی کران‌دار پذیرفته می‌شوند(شکل ۹ را ببینید)

http://dx.doi.org/10.22108/msci.2021.125480.1393



شکل ۹: زبان‌های نوع ۱ از دسته‌بندی چامسکی

۶. نتیجه‌گیری

ابزار مناسب برای زندگی راحت، با توسعه مباحث علوم کامپیوتر ایجاد و بهبود می‌یابند. دست‌یابی به هر توسیعی در این زمینه، آشنایی با مباحث اولیه و پایه در آن را می‌طلبد. در این مقاله چهار مفهوم کلیدی از نظریه محاسبه به طور کامل و خلاصه بحث شده است که می‌توان بخشی از مباحث بیان شده را در جدول ۱۰ خلاصه کرد:

دسته‌بندی چامسکی		
نوع زبان	اتوماتون پذیرنده	گرامر تولیدکننده
نوع 0	ماشین تورینگ	گرامر نامقید
نوع 1	اتوماتون کران‌دار خطی	گرامر حساس به متن
نوع 2	اتوماتون پشته‌ای	گرامر مستقل از متن
نوع 3	اتوماتون متناهی	گرامر منظم

شکل ۱۰: دسته‌بندی چامسکی

مراجع

[1] J. Faizan, *Techniques for Context-Free Grammar Induction and Applications: Application of novel inference algorithms to software maintenance problems*, Germany, VDM Verlag Dr. Müller, 2009.

[2] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[3] P. Linz, *Introduction to Formal Languages and Automata*, 5th edition, Canada, Jones Bartlett Learning, 2011.

[4] J. C. Martin, *Introduction to Languages and The Theory of Computation*, 4th edition, New York, McGraw-Hill Education, 2010.

http://dx.doi.org/10.22108/msci.2021.125480.1393

[5] M. Sipser, *Introduction to the Theory of Computation*, 3rd edition, United States, Cengage Learning, 2012.

[۶] س. تاری، کاربردهایی از اتوماتون متناهی قطعی و غیرقطعی، *نشریه ریاضی و جامعه*، ۴، شماره ۴، (۱۳۹۸).

سمیه تاری
گروه ریاضی، دانشکده علوم پایه، دانشگاه شهید مدنی آذربایجان، تبریز، ایران
s_tari@azaruniv.ac.ir

http://dx.doi.org/10.22108/msci.2021.125480.1393