درس مبانی نظریه محاسبه

جلسه نوزدهم

مسئله تصمیم پذیری

Decideability

تصمیم پذیری

یادآوری: زبان A را قابل تشخیص با تورینگ (یا شمارش پذیر بازگشتی) گوییم اگر ماشین تورینگ M وجود داشته باشد که $L(M) = A$.

تعریف: ماشین تورینگ M را یک تصمیم گیرنده decider گوییم اگر ماشین M برای هر رشته ورودی متوقف شود.

تعریف: زبان A را یک زبان تصمیم پذیر decidable گوییم اگر یک ماشین تورینگ تصمیم گیرنده برای A وجود داشته باشد.

زبانهای تصمیم پذیر در واقع همان مسائل قابل حل هستند. به عبارت دیگر زبانهای تصمیم پذیر مسائلی هستند که برای آنها الگوریتم محاسباتی وجود دارد.

چند پرسش در مورد تصمیم پذیری

پرسش اول: چه مسائلی تصمیم پذیر هستند؟

ما دوست داریم برای مسائل راه حلهای الگوریتمی پیدا کنیم چون با داشتن الگوریتم برای یک مسئله می‌توانیم جواب آن را بصورت مکانیکی بدست آوریم. علاوه بر این، الگوریتمها شناخت ما را در مورد مسائل افزایش می‌دهند.

پرسش دوم: آیا مسئله یا مسائلی وجود دارند که تصمیم پذیر نیستند؟ به عبارت دیگر آیا مسائلی وجود دارند که حل الگوریتمی برای آنها نداریم؟

پرسش سوم: با فرض اینکه مسائل تصمیم ناپذیر وجود دارند، آیا مسائلی ملموس و کاربردی هم وجود دارند که تصمیم پذیر نباشند؟

در این جلسه در مورد پرسش اول بحث می‌کنیم. بحث را با مسئله تصمیم پذیری در مورد خود زبانها و ماشینها شروع می‌کنیم.

تصمیم پذیری زبانها

همانطور که در شروع این درس گفتیم یکی از مسائل مطرح در نظریه محاسبه این است که توصیف زبان A و رشته w را به ما بدهند و بپرسند که آیا رشته w عضو زبان A است یا نه؟

این مسئله کاربرهای فراوان دارد. یکی از کابردهای مهم این مسئله، در طراحی کامپایلرها و زبانهای برنامه‌نویسی است. می‌خواهیم بدانیم آیا کدی که نوشته شده طبق قواعد زبان است یا نه. به عبارت دیگر آیا کد نوشته شده معتبر است یا نه.

اما زبان مورد نظر چگونه بیان شده است؟

چند راه برای توصیف زبانها:

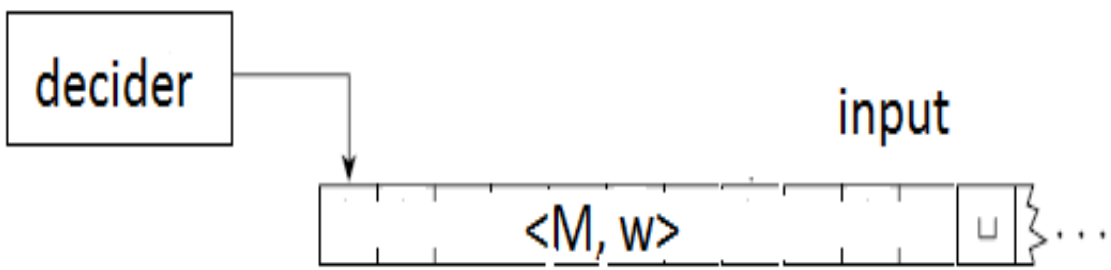
- عبارات منظم
- ماشین
- گرامر

مسئله پذیرش

توصیف ماشین M و رشته w داده شده است. آیا الگوریتمی وجود دارد (به عبارت دیگر، آیا یک ماشین تورینگ تصمیم گیرنده وجود دارد) که مشخص کند رشته w توسط ماشین M پذیرفته می‌شود یا نه.

زبان زیر در این رابطه تعریف شده است.

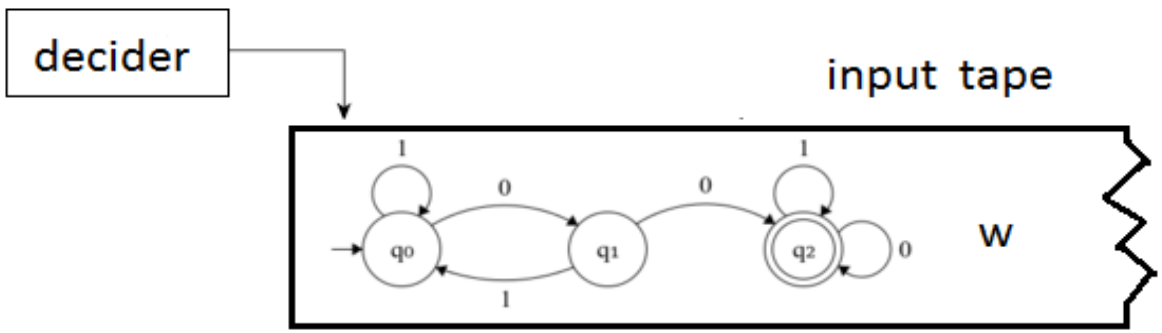
$$\{\langle M, w\rangle \mid M \text{ یک ماشین است که رشته } w \text{ را می‌پذیرد}\}$$



مسئله پذیرش را برای مدلهای مختلف محاسباتی مورد بررسی قرار می‌دهیم.

مسئله پذیرش برای ماشینهای متناهی

$A_{\text{DFA}} = \{\langle B, w\rangle \mid B$ یک ماشین متناهی است که رشته w را می‌پذیرد$\}$



آیا dfa داده شده رشته w را می‌پذیرد؟

راه حل این مسئله روشن است. اجرای ماشین متناهی B را روی رشته w شبیه‌سازی می‌کنیم. اگر در انتهای رشته، ماشین در وضعیت پذیرش بود، ورودی $< B, w >$ را می‌پذیریم در غیر اینصورت رد می‌کنیم.

نتیجه: A_{DFA} یک زبان تصمیم پذیر است.

آیا A_{NFA} هم یک زبان تصمیم پذیر است؟

$A_{\text{NFA}} = \{\langle N, w\rangle | \; N$ ماشین متناهی غیرقطعی است که رشته w را می‌پذیرد$\}$

چون ماشین N غیرقطعی است راه حل اجرای مستقیم مسئله‌دار است. در واقع برای رشته w بایستی همه مسیرهای مختلف از وضعیت شروع را امتحان کنیم.

یک راه حل بهتر این است که اول ماشین N را به یک dfa تبدیل کنیم و سپس dfa بدست آمده را روی رشته w اجرا کنیم.

نتیجه: A_{NFA} یک زبان تصمیم پذیر است.

آزمون تهی بودن زبان یک ماشین

ماشین متناهی A داده شده. آیا زبان A تهی است؟

$$E_{\mathrm{DFA}} = \{\langle A\rangle \mid A \text{ is a DFA and } L(A) = \emptyset\}.$$

ماشین A زبانش تهی است اگر و فقط اگر هیچ مسیری از وضعیت شروع به یک وضعیت پذیرش در ماشین A نداشته باشیم.

این همان مسئله وجود مسیر بین دو راس در یک گراف جهت دار است. این مسئله یک راه حل الگوریتمی دارد (BFS / DFS)

نتیجه: E_{DFA} یک زبان تصمیم پذیر است.

آزمون معادل بودن دو ماشین

ماشینهای متناهی A و B داده شده اند. آیا زبان A و زبان B یکسان هستند؟

$$EQ_{\mathsf{DFA}} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}.$$

ایده حل مسئله:

$$L(A) = L(B) \quad \Leftrightarrow \quad (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)) = \emptyset$$

از روی توصیف ماشینهای A و B یک ماشین برای زبان

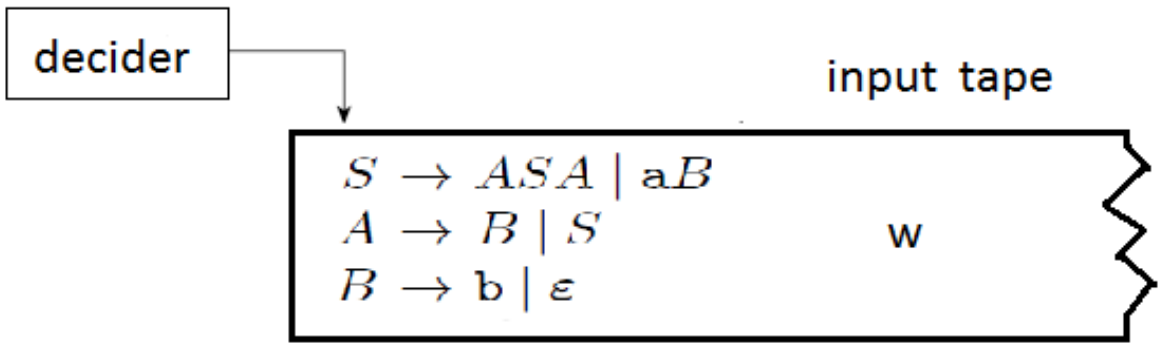
$$(L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$$

می‌سازیم. سپس چک می‌کنیم آیا زبان ماشین مورد نظر تهی است یا نه.

نتیجه: EQ_{DFA} یک زبان تصمیم پذیر است.

مسئله پذیرش برای زبانهای مستقل از متن

$A_{\text{CFG}} = \{\langle G, w\rangle | G$ یک گرامر مستقل از متن است که رشته w را تولید می‌کند$\}$



مسئله: آیا زبان A_{CFG} تصمیم پذیر است؟

اگر بخواهیم همه اشتقاقهای ممکن را چک کنیم، این امکان پذیر نیست چون تعداد اشتقاقهای مختلف بی‌نهایت است و نمی‌توانیم همه را چک کنیم.

کلید حل مسئله: از فرم نرمال چامسکی استفاده می‌کنیم.

حل مسئله: ابتدا گرامر G را به گرامر معادل G' در فرم نرمال چامسکی تبدیل می‌کنیم. در فرم نرمال چامسکی نیازی نیست همه اشتقاقهای ممکن را تست کنیم. فرض کنید طول رشته ورودی n باشد. می‌دانیم اگر رشته w با گرامر نرمال چامسکی G' تولید شود، رشته w با شروع از متغیر S دقیقا بعد از $2n-1$ قدم بدست می‌آید (چرا؟)

پس کافی است همه اشتقاقهای ممکن با تعداد جایگذاری $2n-1$ را چک کنیم. اگر رشته w با هیچ کدام از این اشتقاقها بدست نیامد آنگاه مطمئن هستیم که w توسط گرامر G تولید نمی‌شود.

نتیجه: A_{CFG} یک زبان تصمیم پذیر است.

آزمون تهی بودن زبان یک گرامر

گرامر مستقل از متن G داده شده. آیا زبان G تهی است؟

$$E_{\mathsf{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) = \emptyset\}.$$

زبان گرامر G تهی نیست اگر و فقط اگر متغیر شروع S منجر به تولید یک رشته (حاوی صرفا ترمینال) می‌شود.

فرض کنید B متغیری در گرامر G باشد. آیا B منجر به تولید یک رشته می‌شود؟ این سوال را می‌توانیم در مورد همه متغیرهای گرامر بپرسیم.

الگوریتم پیشنهادی: قوانین گرامر G را در نظر بگیر. همه ترمینالهای قوانین را علامت بزن. آیا متغیری هست که در سمت چپ یک قانون باشد و طرف راست همه علامت خورده باشند؟ اگر چنین متغیری هست همه رخدادهای آن را علامت بزن. این کار را اینقدر تکرار کن تا جایی که همه چیز علامت خورده باشد یا اینکه به بن بست برسیم. در انتها اگر متغیر S علامت نخورده باشد، زبان گرامر تهی است.

نتیجه: E_{CFG} یک زبان تصمیم پذیر است.

مثال: آیا زبان گرامر زیر تهی است؟

$$\begin{array}{l} S \to ASA \mid \mathtt{a}B \\ A \to B \mid S \\ B \to \mathtt{b} \mid \varepsilon \end{array} \Rightarrow \begin{array}{l} S \to ASA \mid \mathtt{a}B \\ A \to B \mid S \\ B \to \mathtt{b} \mid \varepsilon \end{array} \Rightarrow \begin{array}{l} S \to ASA \mid \mathtt{a}B \\ A \to B \mid S \\ B \to \mathtt{b} \mid \varepsilon \end{array} \Rightarrow \begin{array}{l} S \to ASA \mid \mathtt{a}B \\ A \to B \mid S \\ B \to \mathtt{b} \mid \varepsilon \end{array}$$

چون متغیر شروع S علامت خورده است پس گرامر بالا زبانش تهی نیست.

آزمون معادل بودن دو گرامر مستقل از متن

گرامرهای مستقل از متن G و H داده شده اند. آیا زبان G و زبان H یکسان هستند؟

$$EQ_{\text{CFG}} = \{\langle G, H\rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}.$$

شاید بتوانیم مانند ماشینهای متناهی از ایده زیر استفاده کنیم.

$$L(A) = L(B) \quad \Leftrightarrow \quad (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B)) = \emptyset$$

اما یک مشکل وجود دارد. زبانهای مستقل از متن تحت متمم و اشتراک بسته نیستند.

در واقع زبان EQ_{CFG} یک زبان تصمیم پذیر نیست. یعنی ثابت شده هیچ الگوریتمی برای حل این مسئله وجود ندارد.

قضیه: EQ_{CFG} یک زبان تصمیم ناپذیر است.

اثبات این قضیه امری ساده نیست و نیاز به مقدمات بسیار دارد. در واقع ما قضیه بالا را در این کلاس ثابت نمی‌کنیم ولی نشان می‌دهیم زبانهای دیگری هستند که تصمیم پذیر نیستند. این کار را در جلسه آینده انجام می‌دهیم.