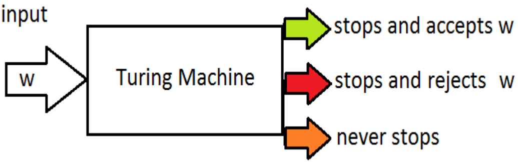
درس مبانی نظریه محاسبه

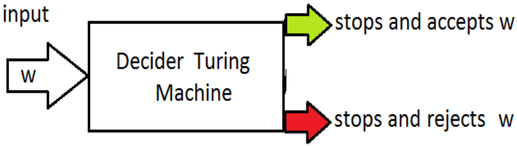
جلسه بیستم

یک مسئله تصمیم ناپذیر

An undecidable problem



ماشین تورینگ



ماشین تورینگ تصمیم گیرنده
(الگوریتم)

یادآوری: زبان A را یک زبان قابل تشخیص با تورینگ Turing-Reconizable گویند اگر ماشین تورینگ M وجود داشته باشد که زبان A را تشخیص دهد.

یادآوری: زبان A را یک زبان تصمیم پذیر decidable گوییم اگر یک ماشین تورینگ برای تشخیص A وجود داشته باشد که برای هر ورودی متوقف شود.

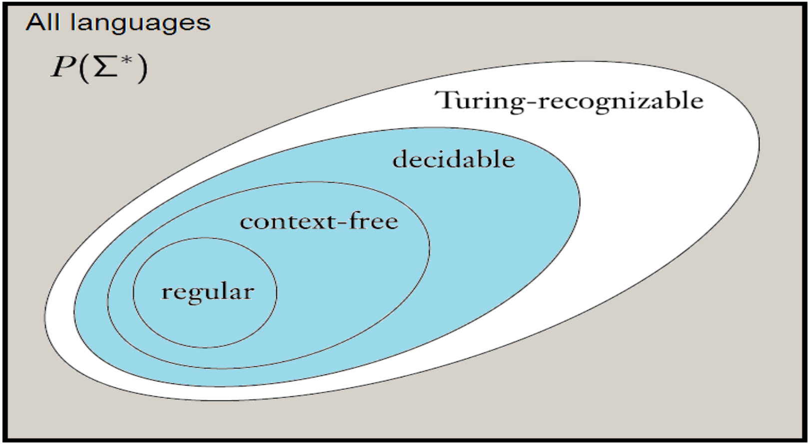
یادآوری: زبان A را یک زبان مستقل از متن context–free گوییم اگر یک ماشین پشته‌ای برای تشخیص A وجود داشته باشد.

یادآوری: زبان A را یک زبان منظم regular گوییم اگر یک ماشین متناهی برای تشخیص A وجود داشته باشد.

سلسله مراتب زبانها

همه زبانها (همه زیر مجموعه‌های Σ^*) $\leftarrow$ $P(\Sigma^*)$

regular $\subset$ context–free $\subset$ decidable $\subset$ Turing–recognizable $\subset P(\Sigma^*)$



موضوع این جلسه

- نشان می‌دهیم بی‌نهایت زبان وجود دارند که قابل تشخیص با تورینگ نیستند (قسمت خاکستری رنگ شکل اسلاید قابل شامل بی نهایت زبان است.) این مطلب از دو گزاره زیر نتیجه می‌شود.

لم 1: مجموعه Turing-recognizable یک مجموعه شماراست.

لم 2: مجموعه همه زبانها $P(\Sigma^*)$ یک مجموعه ناشماراست.

- نشان دهیم زبانی هست که قابل تشخیص با تورینگ است اما تصمیم پذیر نیست. (به عبارت دیگر، می‌خواهیم مسئله‌ای معرفی کنیم که الگوریتمی برای حل آن وجود ندارد.) با توجه به شکل اسلاید قبل، می‌خواهیم زبانی را معرفی کنیم که در قسمت سفید رنگ قرار گرفته است.

قضیه: $\exists A \in$ Turing-recongnizable / decidable

زبان $A \in P(\Sigma^*)$ را در نظر بگیرید. فرض کنید ماشین تورینگ M زبان A را تشخیص دهد. ماشین تورینگ

$$M = (Q, \Sigma, \Gamma, \delta, q_{\text{accept}}, q_{\text{reject}}, q_0)$$

را می‌توان با یک رشته متناهی توصیف کرد زیرا مجموعه‌های Q و Σ و Γ مجموعه‌های متناهی هستند.

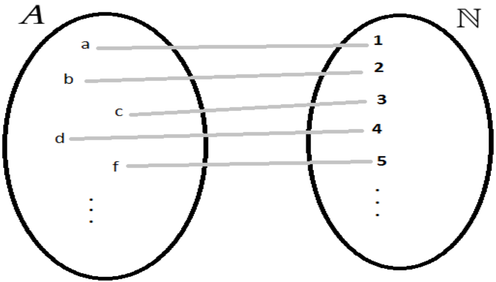
فرض کنید ماشینهای تورینگ برای زبانهای داخل $P(\Sigma^*)$ را با رشته‌هایی از الفبای متناهی Θ توصیف کرده‌ایم. پس $M \in \Theta^*$

فرض کنید $T(\Sigma)$ مجموعه همه ماشینهای تورینگ برای زبانهای داخل $P(\Sigma^*)$ باشد که با الفبای Θ توصیف شده‌اند.

$$T(\Sigma) = \{M_1, M_2, \ldots, \}$$

لم 3: مجموعه ماشینهای تورینگ $T(\Sigma)$ یک مجموعه شماراست.

تعریف: مجموعه A شمارا است اگر یک تناظر یک به یک میان A و زیرمجموعه‌ای از اعداد طبیعی $\mathbb{N}$ وجود داشته باشد.



مثال: مجموعه اعداد گویا شماراست.

مثال: مجموعه اعداد حقیقی شمارا نیست.

مثال: مجموعه $P(\mathbb{N})$ (همه زیرمجموعه‌های $\mathbb{N}$) شمارا نیست.

اثبات لم 3: مجموعه Θ^* را در نظر بگیرید. طبق تعریف $T(\Sigma)$ باید داشته باشیم

$$T(\Sigma) \subset \Theta^*$$

زیرا هر ماشین تورینگ $M \in T(\Sigma)$ رشته‌ای در مجموعه Θ^* است.

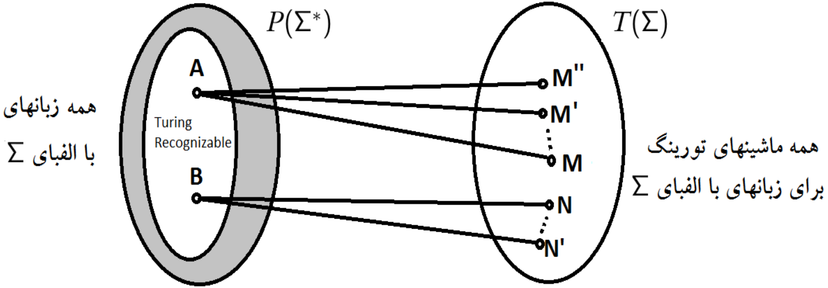
مجموعه Θ^* یک مجموعه شماراست زیرا مجموعه همه رشته‌هایی که از الفبای Θ قابل تولید است را می‌توانیم به ترتیب (ترتیب لغتنامه‌ای استاندارد) بنویسیم. برای مثال فرض کنید $\Theta = \{a, b, c\}$ آنگاه یک ترتیب لغتنامه‌ای استاندارد برای Θ^* بصورت زیر است.

$$a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, \ldots$$

چون $T(\Sigma) \subset \Theta^*$ پس $T(\Sigma)$ هم یک مجموعه شمارا خواهد بود.

لم 1 : مجموعه زبانهای Turing-recognizable یک مجموعه شماراست.

اثبات: هر ماشین تورینگ $M \in T(\Sigma)$ یک زبان در $P(\Sigma^*)$ را تشخیص می‌دهد. پس هر ماشین تورینگ M معادل با یک زبان در $P(\Sigma^*)$ است.

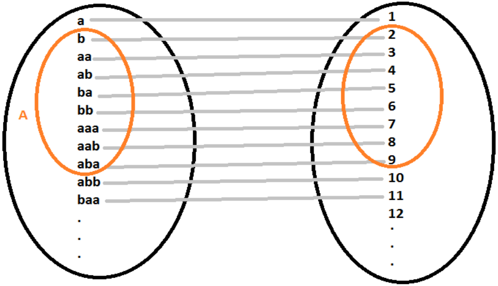


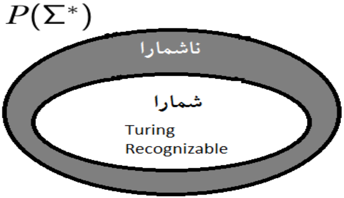
در نتیجه چون $T(\Sigma)$ شماراست پس مجموعه زبانهای Turing–recognizable نیز شماراست.

لم 2: $P(\Sigma^*)$ یک مجموعه ناشمارا است.

اثبات: نشان می‌دهیم یک تناظر یک به یک میان $P(\Sigma^*)$ و $P(\mathbb{N})$ وجود دارد. چون $P(\mathbb{N})$ ناشماراست پس $P(\Sigma^*)$ هم ناشمارا خواهد بود.

چون Σ^* شمارا است (ترتیب لغتنامه‌ای)، پس به هر رشته در Σ^* یک عدد طبیعی منحصربفرد می‌توان نسبت دهیم. هر عضو $P(\Sigma^*)$ یک زیرمجموعه از Σ^* است. پس می‌توان گفت هر عضو $P(\Sigma^*)$ متناظر با یک زیرمجموعه از اعداد طبیعی $\mathbb{N}$ است.





نتیجه: بی نهایت زبان وجود دارد که قابل تشخیص با تورینگ نیستند.

دو نکته:

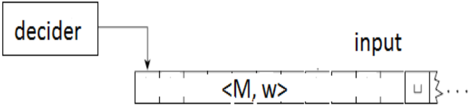
- اثبات بالا وابستگی چندانی به ساختار ماشین تورینگ ندارد. یعنی اگر ما مدل محاسباتی را عوض کنیم باز هم بی‌نهایت زبان وجود خواهد داشت که توسط مدل محاسباتی مورد نظر قابل تشخیص نیستند. تنها شرطش این است که ماشینهای آن مدل را بتوان با یک رشته متناهی توصیف کرد.

- از زبانهایی که قابل تشخیص با تورینگ نیستند، آیا زبانی وجود دارد که توصیف کوتاه و قابل فهمی داشته باشد؟

مسئله پذیرش برای ماشینهای تورینگ

در جلسه گذشته زبان زیر را برای حالی که ماشین داده شده از نوع DFA و NFA بود بررسی کردیم. دیدیم که، در این دو حالت، زبان زیر یک زبان تصمیم پذیر خواهد بود. (برای حالتی که ورودی مسئله یک گرامر مستقل از متن نیز بود، نشان دادیم که زبان مورد نظر تصمیم پذیر است.)

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



اما برای حالتی که ماشین M یک ماشین تورینگ باشد چه؟

$$A_{\mathrm{TM}} = \{\langle M, w\rangle \mid \text{یک ماشین تورینگ است که رشته } w \text{ را می‌پذیرد } M\}$$

زبان A_{TM} تصمیم ناپذیر است

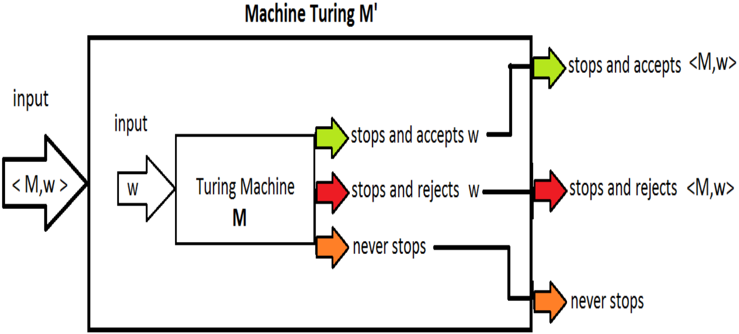
قبل از اینکه اثبات کنیم A_{TM} تصمیم ناپذیر است، ابتدا نشان می‌دهیم که این زبان قابل تشخیص با تورینگ است.

لم 4: $A_{\text{TM}} \in \text{Turing-recognizable}$

اثبات: ماشین تورینگ M' ابتدا توصیف M را می‌خواند و سپس اجرای M را روی رشته w شبیه‌سازی می‌کند. اگر ماشین M متوقف شد و رشته w را پذیرفت، ماشین M' هم متوقف می‌شود به وضعیت q_{accept} می‌رود. اگر M متوقف شد و رشته w را رد کرد، ماشین M' هم متوقف می‌شود و به وضعیت q_{reject} می‌رود. اگر M هیچگاه متوقف نشود، M' هم به طبع آن هیچگاه متوقف نخواهد شد.

ورودی ماشین M' زوج رشته w و توصیف ماشین M است.

ماشین M' اجرای M روی رشته w را شبیه‌سازی می‌کند.



به ماشین تورینگ M' که توصیف یک ماشین تورینگ را به عنوان ورودی دریافت می‌کند و می‌تواند آن را اجرا کند، یک ماشین تورینگ یونیورسال Universal Turnig Machine گفته می‌شود. برنامه سیستم عامل Operating System نوعی از یک ماشین تورینگ یونیورسال است.

قضیه: زبان $A_{\rm TM}$ تصمیم ناپذیر است.

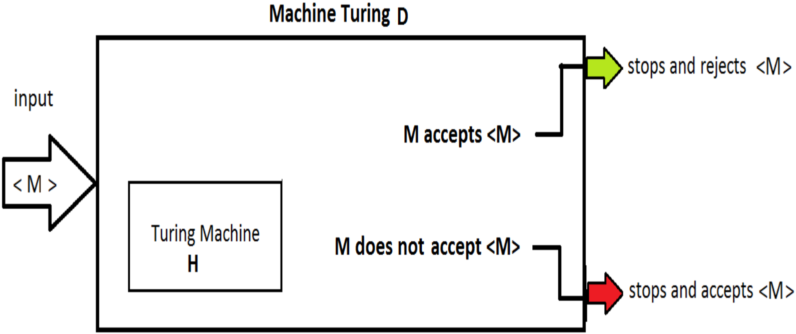
اثبات: بیایید فرض کنیم $A_{\rm TM}$ تصمیم پذیر است. نشان می‌دهیم با این فرض به تناقض می‌رسیم (اثبات با برهان خلف).

فرض کنید H یک ماشین تصمیم گیرنده برای زبان $A_{\rm TM}$ باشد.

$$H(\langle M, w\rangle) = \begin{cases} \text{stops and accepts} & \text{if } M \text{ accepts } w \\ \text{stops and rejects} & \text{if } M \text{ does not accept } w. \end{cases}$$

با استفاده از H ماشین تورینگ تصمیم گیرنده D را می‌سازیم. ماشین D برای رشته ورودی توصیف یک ماشین تورینگ (مثلا $\langle M\rangle$) را دریافت می‌کند. اگر M توصیف خودش را قبول کرد، ماشین D به وضعیت $q_{\rm reject}$ می‌رود و اگر M توصیف خودش را قبول نکرد، ماشین D به $q_{\rm accept}$ می‌رود. ماشین D انگار برعکس M عمل می‌کند.

ماشین تورینگ D، با فرض اینکه ماشین H وجود دارد، یک ماشین تصمیم گیرنده است.



$$D(\langle M\rangle) = \begin{cases} accept & \text{if } M \text{ does not accept } \langle M\rangle \\ reject & \text{if } M \text{ accepts } \langle M\rangle. \end{cases}$$

ماشین D روی توصیف خودش $\langle D\rangle$ باید بصورت زیر عمل کند.

$$D(\langle D\rangle) = \begin{cases} \textit{accept} & \text{if } D \text{ does not accept } \langle D\rangle \\ \textit{reject} & \text{if } D \text{ accepts } \langle D\rangle . \end{cases}$$

اما این یک تناقض است. اگر D توصیف خودش را رد کند باید آن را قبول کند!) به همین ترتیب اگر D توصیف خودش را قبول کند، باید آن را رد کند!)

پس فرض ما مبنی بر وجود ماشین تصمیم گیرنده H برای زبان A_{TM} باید نادرست باشد. در نتیجه A_{TM} تصمیم ناپذیر است. $\square$