دانشکده‌ی علوم ریاضی

نظریه‌ی زبان‌ها و اتوماتا ۲۸ آذر ۱۳۹۱

جلسه‌ی ۲۵: ماشین تورینگ

مدرّس: دکتر شهرام خزائی *نگارندگان: لعیا قدرتی و ستایش ایجادی*

یادآوری

تعریف ۱ *ماشین تورینگ یک هفت تایی منظم است به فرم* $M = (Q, \Sigma, \Gamma, \delta, q_\circ, B, F)$ *که در آن*

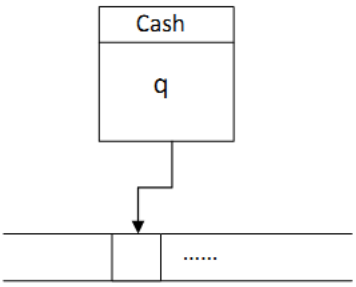
- Q *عبارت است از مجموعه‌ی حالات ماشین.*
- Σ *عبارت است از الفبای رشته ورودی.*
- Γ *عبارت است از الفبای نوار حافظه* (که $\Sigma \subseteq \Gamma$).
- δ *تابع جزئی انتقال حالت است که* $\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, L\}$.
- $q_\circ$ *که همان حالت اولیه است.*
- B *نماد خالی* (مخففBlank) *است که* $B \in \Gamma - \Sigma$.
- F *که مجموعه حالات نهایی است.*

۱ تز چرچ-تورینگ

تز چرچ-تورینگ[۱] بیان می‌کند که هر محاسبه‌ای که توسط ماشین‌های مکانیکی قابل انجام باشد توسط ماشین تورینگ هم قابل انجام است.

نتیجه. اگر تز تورینگ را بپذیریم انتظار داریم که پیچیده تر کردن ماشین تورینگ استاندارد و برای مثال در نظر گرفتن دستگاه حافظه پیچیده تر برای آن ، تاثیری در قدرت ماشین نداشته باشد.

[۱] Charch-Turing thesis



مثال ۱ *انتظار داریم که اضافه کردن حافظه داخلی[۲] به ماشین تورینگ باعث قوی‌تر شدن ماشین نشود.*

۲ ماشین تورینگ با حافظه نهان

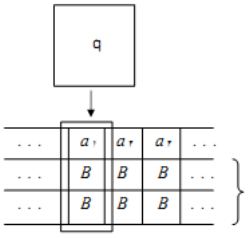
در ماشین تورینگ با حافظه نهان[۳]، حافظه افرایش می‌یابد اما در حقیقت تنها مجموعه حالت‌ها را تغییر می‌دهد. بنابراین این ماشین نیز یک ماشین تورینگ است که مجموعه حالت‌های آن $Q \times \Gamma^k$ است و بقیه‌ی عناصر هیچ تغییری نمی‌کند.

$$\text{جدید } \delta : (Q \times \Gamma^k) \times \Gamma \to (Q \times \Gamma^k) \times \Gamma \times \{L, R\}$$

پس این نوع ماشین، با ماشین تورینگ هم قدرت است اما در بعضی مسائل ممکن است طراحی را ساده‌تر کند.

۳ ماشین تورینگ چندشیاره

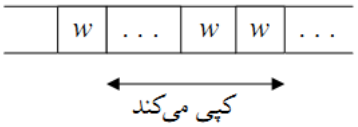
ماشین تورینگ چندشیاره[۴] ماشینی است که هر نوار آن شامل تعدادی شیار است.// در واقع حافظه‌ی ما چند لایه است و در هر دفعه تابع δ روی یک شیار اعمال می‌شود.



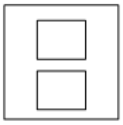
بنابراین الفبای ورودی ما $\Sigma \times \{B^{k-1}\}$ است و الفبای نوار ما Γ^k است.
این ماشین هم باز هم قدرت و معادل ماشین استاندارد است اما گاهی برای حل مساله کاربردی‌تر است.
گاهی می‌خواهیم یک رشته را دنبال کنیم در این صورت یک مارکر[۵] می‌گذاریم و رشته را دنبال می‌کنیم.

[۲] cache
[۳] Turing wachine with cache memory
[۴] multi-track Turing machine
[۵] marker

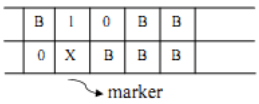
مثال ۲ *ماشین تورینگی طراحی کنید که رشته‌ی ورودی‌اش را تا بی‌نهایت کپی می‌کند.*



جواب. یک ماشین تورینگ درنظر می‌گیریم که یک حالت دارد و یک cache (که یک مقدار می‌گیرد) نوار هد ما دو شیار دارد.



و فرض می‌کنیم رشته‌ی ۱۰ قرار داده شده و قرار است آن را کپی کند. از آنجا که مارکر هست ردیف اول را در cache می‌گذارد بعد اولین جایی‌که B در نوار بالایی دید مقدار cache را خالی می‌کند و دوباره برمی‌گردد و جای مارکر را دوباره عوض می‌کند بنابراین داریم:



مجموعه حالات $Q' = \{q, p, r\}$

q: حرف جدید رشته‌ی ورودی را علامت می‌زند (که در حالت شروع هم هست) (در cache هم می‌ریزد)

p: هد به سمت راست حرکت می‌کند و به دنبال B و شیار بالایی می‌گردد.

r: هد به سمت چپ حرکت می‌کند و به دنبال حرف علامت زده شده می‌گردد.

حال داریم: $\Gamma = \{0, 1, B, X\}$

بنابراین درباره‌ی مجموعه حالات ماشین تورینگ ما می‌توان گفت:

$$Q = \{(x, y) \mid x \in Q',\ y \in \Gamma\}$$

به‌عبارت بهتر می‌توان گفت:

$$Q = \{(q, 0), (q, 1), (q, B), (r, 0), (r, 1), (r, B), (p, 0), (p, 1)\}$$

حال اگر بخواهیم Γ را دقیق‌تر و به فرم ماشین تورینگ استاندارد بنویسیم داریم:

$$\Gamma = \{0, 1, B\} \times \{B, X\}$$

نکته. هیچ‌وقت زیر B یک مارک نمی‌گذاریم، این حالات را باید از Γ بالا حذف کنیم. حال تابع انتقال حالت را می‌نویسیم:

$a, b \in \{0, 1\}$

$\delta((q, B), (a, B)) = ((p, a), (a, X), R)$

در حالت شروع هیچ مارکی نداریم.

حالت دوم:

$$\delta([q,a],[b,B]) = ([p,a),(b,B),R)$$

حالت سوم:

$$\delta((q,a),(B,B)) = ((r,B],(a,B),L)$$

حالت چهارم:

$$\delta([r,B],[a,B]) = ((r,B),(a,B),L)$$

حالت پنجم:

$$\delta((r,B),(a,X)) = ((q,B),(a,B),R)$$

دوباره به‌صورت فوق تکرار می‌کنیم.
دیگه چه حالت‌هایی می‌توانیم اضافه کنیم؟!

۴ ماشین تورینگ با قابلیت توقف هد

ماشین تورینگ با قابلیت توقف هد[۶] ماشینی است که در آن تابع δ ماشین تورینگ استاندارد را به صورت زیر تغییر می‌دهیم:
δ یک تابع جزئی از $Q \times \Gamma$ به $Q \times \Gamma \times \{R, L, S\}$.
اگر ماشین استاندارد ما بتواند ماشین جدید را شبیه‌سازی و پیاده‌سازی کند گوییم ماشین جدید و قدیم معادل‌اند.در واقع زبانی که این دو ماشین می پذیرند یکسان است.

تعریف ۲ (شبیه‌سازی) *می‌گوییم ماشین تورینگ M از یک نوع ماشین تورینگ $\hat{\mathrm{M}}$ از نوع دیگری را شبیه‌سازی می‌کند اگر به‌ازای هر دنباله $I_1, I_2, \ldots, I_n$، از توصیف‌های آنی برای ماشین M که یک دنباله $\hat{I}_1, \ldots, \hat{I}_n$ از توصیف‌های آنی برای ماشین $\hat{\mathrm{M}}$ وجود داشته باشد به‌طوری‌که اگر: $I_1 \vdash I_2 \vdash \cdots \vdash I_n$ آنگاه $\hat{I}_1 \vdash^*_{\hat{\mathrm{M}}} \hat{I}_2 \vdash^*_{\hat{\mathrm{M}}} \cdots \vdash^*_{\hat{\mathrm{M}}} \hat{I}_n$ و این شرط را هم داریم که توصیف آنی $\hat{I}_i$ به‌طور منحصربه‌فردی توصیف آنی I_i را تعیین می‌کند.*

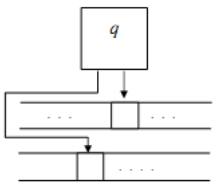
فرض کنیم δ' تابع انتقال حالت ماشین با توقف باشد و δ برای ماشین استاندارد آنگاه

$$\hat{\delta}(q, X) = (p, Y, S)$$

برای پیاده‌سازی آن می‌توانیم یک‌بار به سمت راست برویم و cache را کپی کنیم و به چپ برویم...

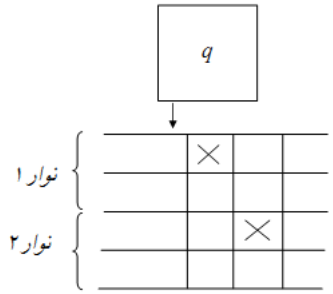
۵ ماشین تورینگ چند نواره

ماشین تورینگ چند نواره[۷] ماشینی دارای چندین نوار حافظه است و به‌ازای هر نوار یک هد مستقل و جداگانه دارد.



[۶] head
[۷] multi-tape

δ یک تابع جزئی از $Q \times \Gamma^k$ به $Q \times \Gamma^k \times \{R, L\}^k$. تعریف می شود، اما همچنان با این قابلیت اضافه ، معادل ماشین استاندارد است.
کافیست ماشین تورینگ جدید را با یک نوار ولی با 2^k
شیار درنظر بگیریم و اثبات کنیم که با ماشین استاندارد معادل است. فرض کنیم که ماشین جدید دو نوار دارد پس ماشین استانداردی با۴ شیار می‌سازیم.



از مارکرها برای مشخص کردن محل هدها استفاده می‌کنیم. مارک‌ها در ردیف اول مربوط به هر نوار قرار دارند.
حال باید تابع δ را به‌طور مناسب تعریف کنیم.
باید با توجه به اینکه چندتا مارکر در سمت راست و چندتا در سمت چپ قرار دارند نتیجه بگیرد که در کدام نوار است.و در هر لحظه می‌داند در کجا قرار دارد و چگونه باید حرکت کند!
با توضیحات بالا ماشین ما می‌تواند در هر لحظه مارکرها را پیدا کند.

۶ ماشین تورینگ غیرقطعی

ماشین تورینگ غیر قطعی (NTM[۸]) ماشینی است که در هر وضعیتی از ماشین برخلاف ماشین تورینگ قطعی می‌تواند مجموعه ای از اعمال مختلف را انجام دهد به این معنا که تابع انتقال حالت آن
δ تغییر می‌کند.
δ_{NTM}: یک تابع جزئی از $Q \times \Gamma$ به $Q \times \Gamma \times \{L, R\}$.

به این معنا که داریم:

$$\delta(p, X) = \{(q_1, X_1, D_1), \ldots, (q_t, X_t, D_t)\}$$

مجاز است که هر کدام از بالایی‌ها را انتخاب کند و اگر نتواند انتخاب کند ماشین تهی است و متوقف می‌شود. در اینجا اصراری نداریم که تابع ما جزئی باشد.

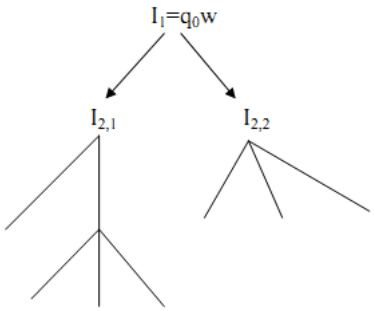
قضیه ۱ *ماشین تورینگ (M_N) که غیرقطعی است را درنظر بگیرید. آنگاه یک ماشین تورینگ قطعی M_D وجود دارد که $L(M_D) = L(M_N)$.*

سوال: ماشین قطعی چگونه می‌تواند ماشین غیرقطعی را شبیه‌سازی کند؟

$$I_1 = q_0 w \vdash I_2 \vdash I_3 \vdash \cdots$$

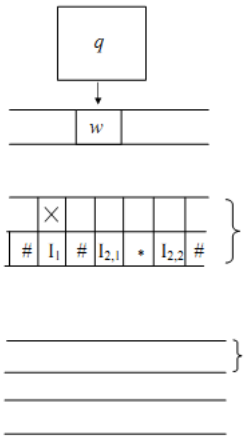
[۸]non-deterministic Turing machine

اگر رشته باعث شود که ماشین به حالت نهایی نرسد یا بی‌نهایت بار ادامه دهد و متوقف نشود آن رشته در زبان نیست. اما در حالت غیرقطعی بسته به تابع انتقال حالت در هر لحظه می‌توان به چند حالت رفت.



اگر در یکی از شاخه‌ها به حالت نهایی رسیدیم ماشین آن مسیر را می‌تواند پیدا کند.
حالت ماشین قطعی ما کافیست search بکند و حالت نهایی را پیدا کند. در هر مرحله حالت ها را ذخیره و چک می‌کند که نهایی هستند یا نه.اگر نهایی نبودند باز الگوریتم را روی هر حالت پیاده خواهد کرد.

ماشین ما بدین شکل است و چند نوار درنظر می‌گیریم. در یکی از نوارها w را می‌خواند در بقیه‌ی نوارها توصیف‌های آن را می‌نویسد.



در هر لحظه همه‌ی توصیف‌های آنی را محاسبه می‌کند و بررسی می‌کند.
زبان ماشین‌های تورینگ را جوری تعریف کردیم که نهایت ماشین متوقف می‌شود و می‌گوید رشته در زبان هست یا نه اما اگر متوقف شود هیچ وقت در زمان متناهی نمی‌توانیم تصمیمی راجع به آن بگیریم یعنی یک سری زبان‌ها هستند که تصمیم پذیر نیستند.
زبانی که ماشین‌های تورینگ می‌پذیرند را می‌گوییم Recursive Enumerable (بازگشتی برشمردن)