

درس مبانی نظریه محاسبه

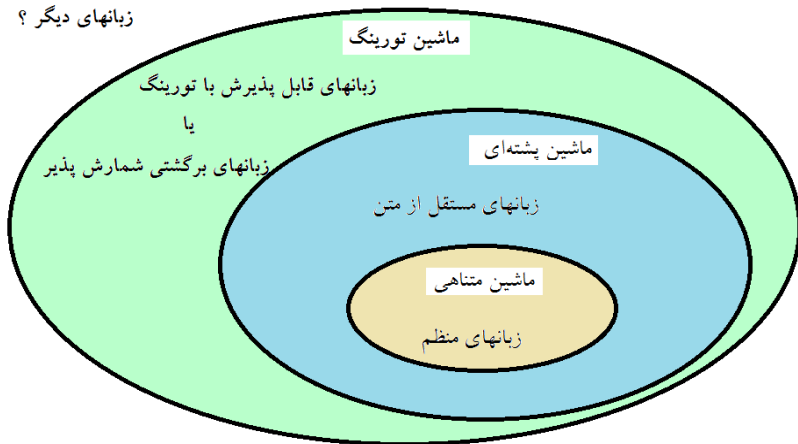
جلسه هجدهم

گونه‌های مختلف ماشین تورینگ، تز چرچ –
تورینگ

Church-Turing Thesis

تعریف: زبان A را یک زبان پذیرش شونده با تورینگ یا برگشتی شمارش پذیر گویند اگر ماشین تورینگ M وجود داشته باشد که زبانش مجموعه A باشد.

زبانهای دیگر ؟



Turing-Recognizable or Recursively Enumerable Languages

آیا می‌توانیم مدلی قدرتمندتر از ماشین تورینگ داشته باشیم؟ یعنی ماشینهایی داشته باشیم که علاوه بر زبانهای برگشتی شمارش پذیر، زبانهای دیگر را نیز تشخیص دهند؟

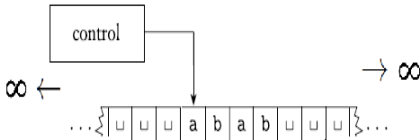
یک ایده برای پیدا کردن یک مدل قوی‌تر، دادن امکانات بیشتر به ماشین تورینگ کلاسیک است.

برای مثال، نوک خواندن/نوشتن در ماشین تورینگ کلاسی در هر قدم باید یا به راست و یا به چپ حرکت کند. اگر علاوه بر چپ و راست رفتن، گزینه ماندن در خانه کنونی را نیز اضافه کنیم، آیا قدرت ماشین تورینگ افزایش می‌یابد؟

به راحتی می‌توان دید که ایجاد چنین گزینه‌ای چیزی به قدرت ماشین تورینگ اضافه نمی‌کند. چون گزینه ماندن در خانه کنونی را می‌توان با دو حرکت راست و چپ **شبه‌سازی** کرد.

نوار نامحدود از دو جهت

شاید یک راه برای قدرت بخشیدن به ماشین تورینگ این باشد که نوار را از دو جهت نامحدود کنیم. مانند شکل زیر.

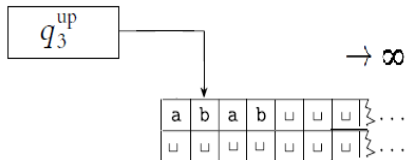


مدل بالا را می‌توان با نوار از یک جهت نامحدود شبیه‌سازی کرد. این کار را در دو مرحله انجام می‌دهیم.

در مرحله اول، می‌توانیم فرض بگیریم که وسط نوار (محل آغازین نوک خواندن/نوشتن) همواره مشخص است. این کار را با علامت زدن خانه وسط انجام می‌دهیم. مثلاً، کاراکتری که آنجا نوشته می‌شود را علامت می‌زنیم. برای نیل به این هدف، الفبای نوار را افزایش می‌دهیم:

$$\Gamma' = \{a, b, \sqcup, a^+, b^+, \sqcup^+\}$$

حال می‌توانیم فرض بگیریم که وسط نوار همواره مشخص است. دو جهت نامحدود را با یک نوار دوبانده یک جهت نامحدود شبیه‌سازی می‌کنیم. در بانده بالا، محتوای سمت راست قرار می‌گیرد و در بانده دوم محتوای سمت چپ نوار قرار می‌گیرد. برای اینکه مشخص شود نوک خواندن/نوشتن در کدام سمت واقع شده، می‌توانیم به تعداد وضعیتهای ماشین اضافه کنیم. مثلاً اگر ماشین در وضعیت q_3^{up} قرار بگیرد یعنی نوک خواندن/نوشتن در بانده بالاست.

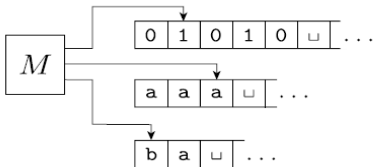


الفبای نوار هم متناسب با تغییرات افزایش می‌یابد.

$$\Gamma'' = \left\{ \frac{a}{b}, \frac{b}{a}, \frac{a}{a}, \frac{\sqcup}{a}, \dots \right\}$$

ماشین تورینگ چندنواره

مانند ماشین تورینگ کلاسیک است با این تفاوت که به جای یک نوار چندین نوار دارد. هر نوار نوک خواندن/نوشتن مخصوص به خود را دارد. در ابتدا، رشته ورودی روی نوار اول قرار می‌گیرد و بقیه نوارها خالی هستند.

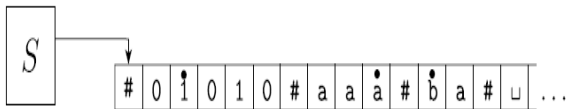


با فرض اینکه ماشین k نوار دارد، تابع تغییر وضعیت δ بصورت زیر تعریف می‌شود.

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, S\}^k$$

در هر قدم، همه نوکهای خواندن/نوشتن می‌توانند حرکت کنند. اینجا S به معنای ماندن در جای خود است.

ماشین تورینگ چندنواره را می‌توان با ماشین تورینگ یک نواره شبیه‌سازی کرد.



- ▶ محتوای نوارها را می‌توان روی یک نوار نوشت و از کاراکتر # برای حائل بین نوارها استفاده کرد.
- ▶ برای اینکه محل نوک خواندن/نوشتن روی هر نوار مشخص باشد، می‌توانیم از ترفند علامت زدن استفاده کنیم. روی هر قسمت، کاراکتری که علامت خورده است، محل قرار گرفتن نوک را نشان می‌دهد.
- ▶ ماشین با رفت و آمد روی نوار، حرکت همزمان همه نوارها در ماشین چندنواره را شبیه‌سازی می‌کند. اگر قسمتی نیاز به گسترش داشت، ماشین با شیفت دادن محتوای نوار، فضای خالی برای قسمت مورد نظر ایجاد می‌کند.

ماشین تورینگ غیرقطعی

Non-deterministic Turing Machine

ماشین تورینگ غیرقطعی همانند ماشین تورینگ کلاسیک است با این تفاوت اساسی که برای قدم بعدی ماشین چند گزینه می‌تواند وجود داشته باشد. به عبارت دیگر تابع تغییر وضعیت بصورت زیر تعریف می‌شود.

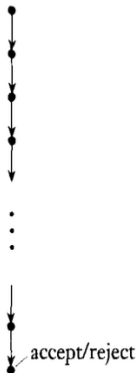
$$\delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$$

برای مثال اگر ماشین در وضعیت q_2 باشد و محتوای زیر نوک خواندن/نوشتن علامت a باشد ماشین دو گزینه دارد: به سمت راست برود و خانه مورد نظر را تغییر ندهد، وضعیت نیز تغییری نکند یا اینکه به سمت چپ برود و خانه مورد نظر را به علامت b تغییر دهد. وضعیت ماشین نیز به q_3 تغییر کند.

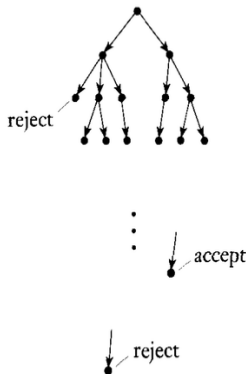
$$\delta(q_2, a) = \{(q_2, a, R), (q_3, b, L)\}$$

ماشین تورینگ غیر قطعی را می توان با ماشین تورینگ کلاسیک شبیه سازی کرد.
 ماشین تورینگ غیر قطعی چه زمانی رشته w را می پذیرد؟ اگر درخت محاسبات
 w برگ با برچسب پذیرش accept داشته باشد. دقت کنید در ماشین قطعی
 فقط یک مسیر وجود دارد (درخت مربوطه یک مسیر ساده است).

Deterministic



Nondeterministic



یک راه برای شبیه‌سازی ماشین قطعی این است که برای یک رشته همه مسیرهای ممکن را امتحان بکنیم. در واقع درخت محاسبات w را می‌گردیم تا اینکه یک برگ با برچسپ پذیرش accept پیدا کنیم.

دو راه برای جستجو در درخت w وجود دارد.

◀ جستجوی عمقی

اشکال این روش این است که ممکن است مسیری از ریشه به پایین تا بی‌نهایت ادامه پیدا کند. همانطور که می‌دانیم ماشین تورینگ ممکن است هرگز متوقف نشود. پس اگر مسیری را بطور عمقی دنبال کنیم شاید هیچگاه به انتهای آن نرسیم و در نتیجه فرصت برای جستجوی بقیه درخت از دست برود (شاید در جایی در درخت برگی با برچسپ accept داشته باشیم و از زیارت آن محروم شویم).

◀ جستجوی سطحی

از این روش برای جستجوی درخت استفاده می‌کنیم.

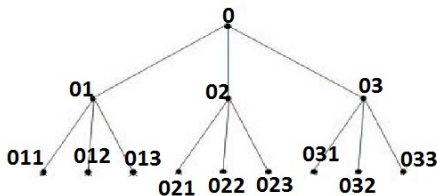
اما چگونه درخت محاسبات w را بصورت سطحی جستجو کنیم؟

اولین کاری که می‌کنیم، رئوس درخت محاسبات را کدگذاری می‌کنیم. یعنی به هر راس درخت یک شماره اختصاص می‌دهیم.

فرض کنید در ماشین غیرقطعی M ، حداکثر تعداد گزینه‌ها برای قدم بعدی b تا باشد. یعنی در هر زمان ماشین حداکثر b گزینه برای قدم بعدی دارد.

پس هر راس غیربرگ درخت محاسبات w ، به تعداد b فرزند دارد. (البته ممکن است راسی کمتر از b فرزند داشته باشد. در صورت نیاز می‌توانیم چند فرزند بی‌استفاده به همه رئوس اضافه کنیم تا هر راس غیربرگ، دقیقاً به تعداد b فرزند داشته باشد.)

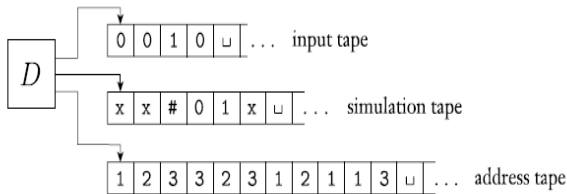
به ریشه درخت را شماره 0 را اختصاص می‌دهیم. فرزندان ریشه به ترتیب از چپ به راست شماره‌های $01, 02, 03, \dots, 0b$ می‌گیرند. فرزند اول 01 (اگر فرزندی باشد) شماره 011 دریافت می‌کند. فرزند دوم 01 شماره 012 دریافت می‌کند. با همین روش کل رئوس را شماره‌گذاری می‌کنیم.



هر راس درخت محاسبه یک پیکربندی ماشین است (که ممکن است یک پیکربندی پذیرش یا عدم پذیرش و یا یک پیکربندی معمولی باشد).

شبیه‌سازی را با یک ماشین سه‌نواره قطعی انجام می‌دهیم. این کار اشکالی ندارد چون از قبل می‌دانیم که ماشین سه‌نواره را می‌توانیم با ماشین یک نواره شبیه‌سازی کنیم.

ماشین سه‌نواره هر بار یکی از رئوس درخت محاسبه W را می‌آزماید. رشته ورودی W روی نوار اول نوشته می‌شود. نوار اول دست نخورده باقی می‌ماند.



از نوار دوم برای شبیه‌سازی ماشین غیرقطعی استفاده می‌کنیم. اینجا مثل چرکنویس عمل می‌کند. محاسبات ماشین اینجا انجام می‌شود.

از نوار سوم برای مشخص کردن آدرس (یا شماره) راس درخت محاسبه هدف استفاده می‌کنیم. مثلاً اگر روی نوار سوم نوشته شده باشد 031 یعنی می‌خواهیم راس با شماره 031 را بررسی کنیم. دقت کنید محتوای شماره ما را به سمت پیکربندی مورد نظر هدایت می‌کند. برای مثال این شماره به ما می‌گوید که در اولین انتخاب گزینه سوم را انتخاب کن. برای انتخاب دوم از گزینه اول استفاده کن. اگر با این انتخابها ماشین به یک پیکربندی پذیرش رسید، آنگاه ماشین متوقف می‌شود و اعلام می‌کند که رشته قبول شده است. در غیر این صورت باید سراغ یک راس دیگر درخت برود.

اگر راس مورد نظر یک پیکربندی پذیرش نباشد، باید سراغ یک راس دیگر برویم. راس بعدی در درخت را انتخاب می‌کنیم و شماره آن را روی نوار سوم می‌نویسیم. حال همه چیز را روی نوار دوم پاک می‌کنیم و دوباره از ریشه شروع کرده و با استفاده از نوار سوم به سمت راس هدف حرکت می‌کنیم.

تز چرچ - تورینگ

هر مدل محاسباتی توسط ماشین تورینگ کلاسیک قابل شبیه‌سازی است.

تز چرچ - تورینگ اثبات نشده است اما بطور عام مورد قبول دانشمندان واقع شده است. تا کنون همه تلاشها برای رد این تز ناموفق بوده است.

با توجه به جامع بودن مدل محاسباتی ماشین تورینگ، از این به بعد از توصیف الگوریتمی و سطح بالا برای ماشین تورینگ استفاده می‌کنیم. در واقع الگوریتم یک توصیف دیگر برای ماشین تورینگ است.