

درس مبانی نظریه محاسبه

جلسه هفدهم

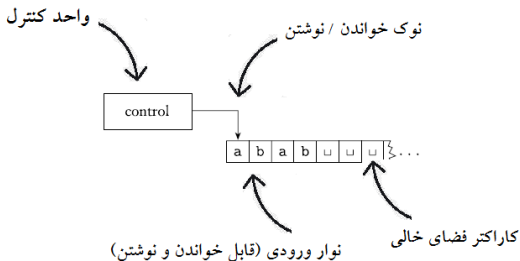
فصل سوم: ماشین تورینگ

Turing Machines

ماشینهای محاسباتی

مدل محاسباتی	تعداد وضعیتهای	حافظه	دسترسی به حافظه
ماشین متناهی	متناهی	—	—————
ماشین پشته‌ای	متناهی	نامتناهی	محدود
ماشین تورینگ	متناهی	نامتناهی	نامحدود

ماشین تورینگ



▶ ماشین تورینگ مانند ماشین پشته‌ای یک نوار ورودی دارد که در آن رشته ورودی قرار می‌گیرد.

▶ ماشین تورینگ برخلاف ماشین پشته‌ای یک بخش مجزا برای حافظه ندارد. در ماشین تورینگ از نوار ورودی به عنوان حافظه استفاده می‌کند. ماشین می‌تواند هم نوار ورودی را بخواند و هم آن را بنویسد.

◀ طول نوار ورودی نامحدود است.

◀ در شروع کار ماشین، رشته ورودی در انتهای سمت چپ نوار قرار گرفته و نوک خواندن/نوشتن روی اولین حرف رشته قرار گرفته است.

◀ نوک خواندن و نوشتن می‌تواند به سمت چپ یا راست حرکت کند و یا اینکه در جای خودش باقی بماند. در صورتی که نوک خواندن/نوشتن در انتهای سمت چپ باشد و ماشین دستور دهد به سمت چپ برود طبق قرارداد نوک در جای خود باقی می‌ماند.

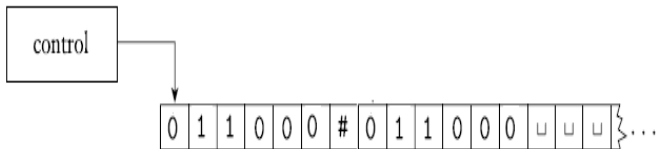
◀ برای مشخص شدن پایان محتوا (در نوار ورودی) از کاراکتر فضای خالی □ استفاده می‌شود. ماشین وقتی به کاراکتر □ رسید به این معنی است که فضای سمت راست خالی است و چیزی آنجا نوشته نشده است.

◀ ماشین می‌تواند محتوای یک خانه از نوار را به کاراکتر □ تغییر دهد.

- ▶ ماشین تورینگ یک وضعیت پذیرش Accpet و یک وضعیت عدم پذیرش Reject دارد. ماشین به محض ورود به وضعیت پذیرش و یا عدم پذیرش متوقف می شود (اگرچه همه رشته ورودی را خوانده باشد).
- ▶ ماشین تنها زمانی رشته ورودی را می پذیرد که متوقف شده باشد و در وضعیت پذیرش Accept قرار گرفته باشد.
- ▶ ماشین تورینگ کلاسیک قطعی است (در هر زمان تنها یک گزینه برای حرکت بعدی وجود دارد).
- ▶ ماشین تورینگ اساس رایانه های امروزی است. هر کاری که رایانه های امروزی انجام دهند مدل ساده ماشین تورینگ هم می تواند انجام دهد (البته با صرف زمان بیشتر!)

یک ماشین تورینگ برای زبان

$$B = \{w\#w \mid w \in 0, 1^*\}$$



توصیف کلی ماشین: ماشین باید روی نوار رفت و برگشت کند و حروف در مکانهای متقارن در دو سوی علامت # را با هم تطبیق دهد. اگر شاهدهی برای عدم تساوی پیدا شد به وضعیت reject می‌رود و متوقف می‌شود در غیر این صورت پس از انجام همه تطبیق‌ها به وضعیت accept می‌رود و متوقف می‌شود.

ترفند خط زدن علامتها

ماشین چگونه دو کاراکتر در دو سوی # را تطبیق می دهد و چه تمهیدی برای این دارد که دوباره یک زوج کاراکتر را تطبیق ندهد؟

برای حل این چالش، ماشین از ترفند خط زدن کاراکترها استفاده می کند. برای این کار ماشین می تواند کاراکترهای 0 و 1 را با علامت جدید x عوض کند تا دوباره کاراکتر مورد نظر تطبیق داده نشود.

0 1 1 0 0 0 # 0 1 1 0 0 0 □ ...
x 1 1 0 0 0 # 0 1 1 0 0 0 □ ...
x 1 1 0 0 0 # x 1 1 0 0 0 □ ...
x 1 1 0 0 0 # x 1 1 0 0 0 □ ...
x x 1 0 0 0 # x 1 1 0 0 0 □ ...
x x x x x x # x x x x x x □ ...
accept

توصیف رسمی یک ماشین تورینگ

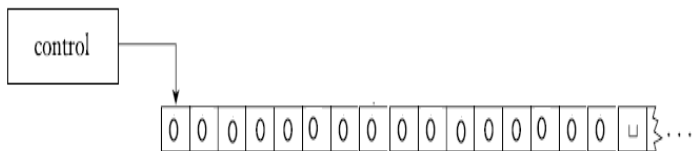
DEFINITION

A *Turing machine* is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$

Q, Σ, Γ are all finite sets

1. Q is the set of states
2. Σ is the input alphabet not containing the *blank symbol* \sqcup
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$ is the transition function
5. $q_0 \in Q$ is the start state
6. $q_{\text{accept}} \in Q$ is the accept state
7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{reject}} \neq q_{\text{accept}}$.

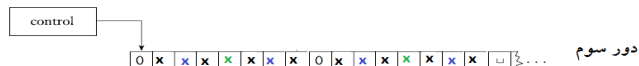
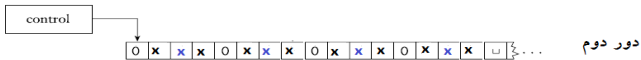
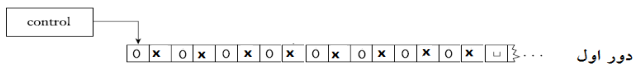
یک ماشین تورینگ برای زبان $A = \{0^{2^n} \mid n \geq 0\}$



توصیف کلی ماشین: ماشین اول چک می کند اگر طول رشته ورودی 1 باشد ورودی را می پذیرد. اگر طول رشته ورودی 0 یا یک عدد فرد باشد (بزرگتر از یک) ماشین ورودی را رد می کند. در غیر اینصورت، ماشین هر بار طول رشته را نصف می کند (تقسیم بر 2) می کند. اگر بعد از تقسیم بر 2 کردنهای متوالی طول رشته حاصل به 1 رسید ماشین به وضعیت accept می رود و متوقف می شود. اگر بعد از یک عمل نصف کردن، طول رشته حاصل 0 و یا یک عدد فرد بزرگتر از 1 شد، ماشین به وضعیت reject می رود و متوقف می شود.

سوال: ماشین چگونه می فهمد طول رشته یک عدد فرد بزرگتر از 1 است؟
برای این کار، مانند یک ماشین متناهی عمل می کند.

سوال: ماشین چگونه طول رشته را نصف می کند؟
برای این کار، از ایده خط زدن استفاده می کنیم. در شروع هر دور، نوک خواندن و نوشتن به ابتدای نوار برمی گردد. ماشین یک در میان کاراکترهای 0 را خط می زند. سپس دوباره به ابتدای نوار برمی گردد و چک می کند آیا تعداد صفرهای باقیمانده یک عدد فرد بزرگتر از 1 است یا نه.

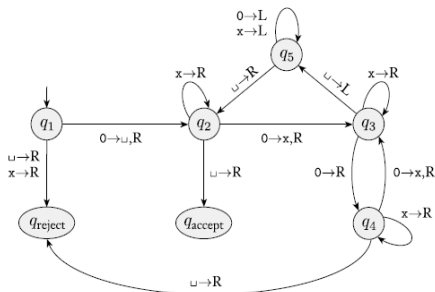


توصیف رسمی ماشین تورینگ M_2 برای زبان

$$A = \{0^{2^n} \mid n \geq 0\}$$

$$M_2 = (Q, \Sigma, \Gamma, \delta, q_1, q_{\text{accept}}, q_{\text{reject}})$$

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{\text{accept}}, q_{\text{reject}}\}$
- $\Sigma = \{0\}$
- $\Gamma = \{0, x, \sqcup\}$

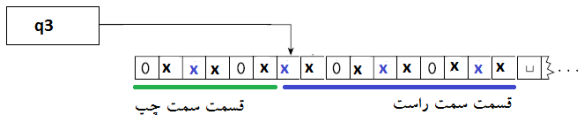


بدلیل نیاز به جزئیات بسیار، در این کلاس از توصیف رسمی ماشین تورینگ صرف نظر می کنیم و ماشین را به شکل کلی و سطح بالا بیان می کنیم.

پیکربندی یک ماشین تورینگ configuration

تعریف: به وضعیت کلی که ماشین در یک زمان مشخص در آن قرار دارد (شامل وضعیت واحد کنترل، محتوای نوار و محل نوک خواندن/نوشتن) یک پیکربندی گفته می‌شود. پیکربندی در واقع تصویر ماشین در یک مقطع زمانی است.

برای مثال، واحد کنترل ماشین زیر در وضعیت q_3 قرار دارد. محتوای نوار را با توجه به محل قرار گرفتن نوک خواندن/نوشتن می‌توانیم به دو قسمت تقسیم کنیم. در مثال زیر، $0xxx0x$ قسمت سمت چپ نوک خواندن/نوشتن است و $xx0xxx0xxx$ قسمت سمت راست شامل کاراکتر زیر نوک می‌باشد.



پیکربندی بالا را بصورت زیر نشان می‌دهیم.

$$0xxx0x \quad q_3 \quad xx0xxx0xxx$$

در آغاز کار، ماشین تورینگ در **پیکربندی شروع** قرار می گیرد. در اینجا w رشته ورودی است.

$$q_0 w \Rightarrow \text{پیکربندی شروع}$$

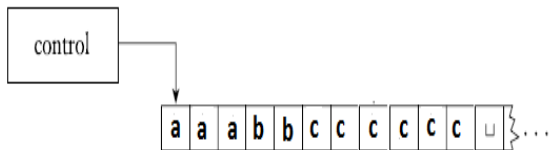
ماشین تورینگ در هر پیکربندی که قرار بگیرد، حرکت بعدی و پیکربندی بعدی از روی توصیف ماشین بدست می آید.

یک ماشین تورینگ با شروع پردازش رشته ورودی یکی از سه سرنوشت زیر را پیدا می کند.

- ▶ ماشین متوقف می شود و در وضعیت q_{accept} قرار می گیرد.
- ▶ ماشین متوقف می شود و در وضعیت q_{reject} قرار می گیرد.
- ▶ ماشین هرگز متوقف نمی شود! در این حالت گوییم که ماشین در دور بینهایت افتاده است.

تعریف: ماشین تورینگ M رشته w را می‌پذیرد اگر و فقط اگر ماشین از پیکربندی شروع $w q_0$ کارش را آغاز کرده و بعد از چند حرکت متناهی متوقف شود و در یک پیکربندی پذیرش قرار بگیرد.

یک ماشین تورینگ برای زبان $A = \{a^i b^j c^k \mid i \times j = k, i, j, k \geq 0\}$



توصیف کلی ماشین: ماشین ابتدا چک می کند که رشته ورودی الگوی $a^*b^*c^*$ را داشته باشد. اگر این الگو را نداشته باشد ورودی را reject می کند. سپس به ابتدای رشته برمی گردد و هر بار یک a را خط می زند و سپس به تعداد b ها از قسمت c خط می زند. در هر مرحله به ابتدای رشته برمی گردد و این کار را انجام می دهد. در هر مرحله اگر عمل مورد نظر با موفقیت انجام نشد رشته ورودی را reject می کند در غیر این صورت در انتها رشته را می پذیرد.