به نام خدا

نظریهٔ زبان‌ها و ماشین‌ها

آرش شفیعی

ماشین‌های تورینگ

مقدمه

- با استفاده از لم تزریق برای زبان‌های مستقل از متن، نشان دادیم که زبان‌هایی مانند $\{a^n b^n c^n\}$ و $\{ww\}$ مستقل از متن نیستند و بنابراین برای شناسایی آنها به ماشین‌های قدرتمندتری نیاز داریم.
- ماشین‌های پشته‌ای به دلیل داشتن پشته نسبت به ماشین‌های متناهی قدرت بیشتر پیدا کردند. می‌توانیم حدس بزنیم که با داشتن حافظه‌ای با انعطاف بیشتر نسبت به پشته می‌توانیم ماشینی بسازیم که از ماشین‌های پشته‌ای قوی‌ترند.
- در این قسمت ماشین‌های تورینگ را معرفی می‌کنیم و با استفاده از این ماشین‌ها مفهوم الگوریتم محاسباتی را تعریف می‌کنیم و در پایان نشان می‌دهیم که این ماشین‌ها هر نوع محاسباتی را می‌توانند انجام دهند.

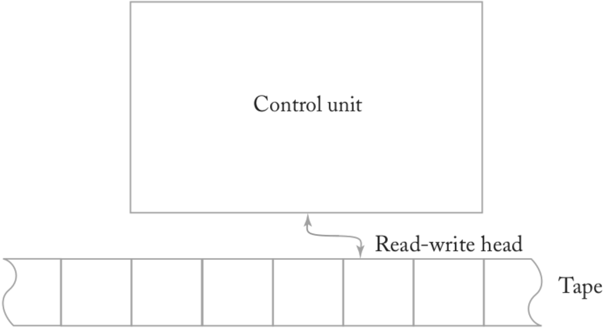
ماشین تورینگ استاندارد

- ماشین تورینگ ماشینی است که حافظه موقت آن یک نوار [1] است.
- این نوار از سلول‌ها [2] (خانه‌ها) پی تشکیل شده است که هر کدام یک نماد را در بر می‌گیرند.
- یک هد (کلاهک) خواندن و نوشتن [3] بر روی نوار قرار گرفته است که قادر است به سمت راست و چپ حرکت کند و در هر حرکت نمادی را از روی یکی از سلول‌های نوار بخواند و یا نمادی را بر روی یک سلول بنویسد.
- بنابراین نوار و هد بر روی آن، مکانیزم ورودی و خروجی این ماشین را تشکیل می‌دهند.

[1] tape
[2] cell
[3] read–write head

ماشین تورینگ استاندارد

- یک ماشین تورینگ را می‌توان بدین شکل نشان داد.



ماشین تورینگ استاندارد

- یک ماشین تورینگ M با یک هفت‌تایی $M = (Q, \Sigma, \Gamma, \delta, q_0, \Box, F)$ تعریف می‌شود، به طوری که:
- Q مجموعهٔ حالات داخلی ماشین است.
- Σ الفبای ورودی است.
- Γ مجموعه‌ای متناهی از نمادهاست به نام الفبای نوار [1].
- δ تابع گذار است.
- $\Box \in \Gamma$ یک نماد ویژه به نام نماد نانوشته [2] است.
- $q_0 \in Q$ حالت آغازی است.
- $F \subseteq Q$ مجموعه‌ای از حالت‌های پایانی است.

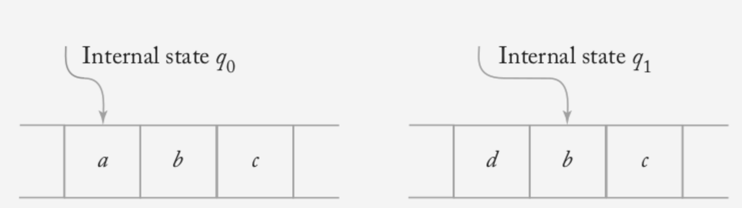
---
[1] tape alphabet
[2] blank

ماشین تورینگ استاندارد

- در ماشین تورینگ الفبای ورودی زیر مجموعه‌ای است از الفبای نوار و نماد نانوشته را در بر نمی‌گیرد. به عبارت دیگر $\Sigma \subseteq \Gamma - \{\Box\}$
- همچنین تابع گذار یک تابع جزئی است که به صورت $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ تعریف می‌شود.
- بنابراین تابع گذار به طوری تعریف می‌شود که ماشین با خواندن یک نماد از نوار بر اساس حالتی که در آن قرار دارد به یک حالت دیگر گذار می‌کند و یک نماد دیگر بر روی نوار می‌نویسد. سپس هد نوار به سلول سمت چپ و یا به سلول سمت راست حرکت می‌کند.

ماشین تورینگ استاندارد

- برای مثال با تابع گذار $\delta(q_0, a) = (q_1, d, R)$ نوار از پیکربندی شکل سمت چپ به پیکربندی شکل سمت راست تغییر می‌کند.



ماشین تورینگ استاندارد

- ماشین تورینگ یک مدل اولیه برای کامپیوترهای امروزی است.
- این ماشین یک واحد پردازش دارد که حافظه محدود دارد و یک حافظهٔ جانبی دارد که از لحاظ نظری نامحدود است.
- تعداد دستورات این ماشین (که یک پردازنده است) بسیار محدود است. این ماشین تنها می‌تواند یک نماد را بخواند و تصمیم بگیرد که به چه حالتی برود، چه نمادی را بنویسد و هد خود را به چه سمتی حرکت دهد.

ماشین تورینگ استاندارد

- به نظر می‌رسد ماشین تورینگ بسیار ساده و ابتدایی باشد ولی می‌تواند عملیات پیچیده‌ای انجام دهد. تابع گذار این ماشین را برنامه [1] ماشین تورینگ می‌گوییم.
- ماشین از حالت آغازی شروع به کار می‌کند، با خواندن نمادها تعدادی سلول نوار را تغییر می‌دهد، و از حالتی به حالت دیگر می‌رود و در پایان در صورتی که در حالتی قرار بگیرد که هیچ گذاری تعریف نشده باشد، به حالت توقف [2] می‌رود.
- در یک حالت توقف هیچ گذاری تعریف نشده است. همچنین در حالت‌های پایانی در ماشین تورینگ هیچ گذاری تعریف نشده است، پس هر گاه یک ماشین توینگ به یک حالت پایانی وارد می‌شود متوقف می‌شود.

[1] program
[2] halt state

ماشین تورینگ استاندارد

- یک ماشین تورینگ را که به صورت $Q = \{q_0, q_1\}, \Sigma = \{a, b\}, \Gamma = \{a, b, \square\}, F = \{q_1\}$ تعریف شده است در نظر بگیرید.

- توابع گذار به صورت زیر هستند.

$$\delta(q_0, a) = (q_0, b, R)$$
$$\delta(q_0, b) = (q_0, b, R)$$
$$\delta(q_0, \square) = (q_1, \square, L)$$

- این ماشین چه عملیاتی انجام می‌دهد؟

ماشین تورینگ استاندارد

- یک ماشین تورینگ را که به صورت $Q = \{q_0, q_1\}, \Sigma = \{a, b\}, \Gamma = \{a, b, \square\}, F = \{q_1\}$ تعریف شده است در نظر بگیرید.

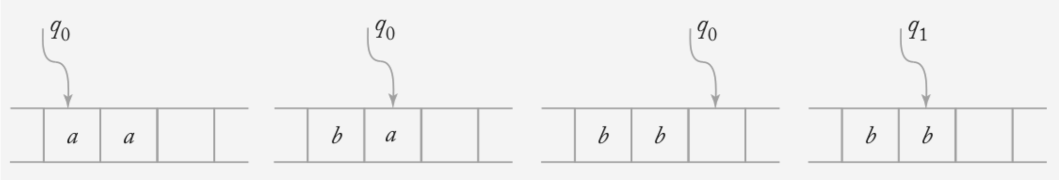
- توابع گذار به صورت زیر هستند.

$$\delta(q_0, a) = (q_0, b, R)$$
$$\delta(q_0, b) = (q_0, b, R)$$
$$\delta(q_0, \square) = (q_1, \square, L)$$

- این ماشین به ازای هر نماد a خوانده شده، آن را به b تغییر می‌دهد و در نهایت با خواندن نماد نانوشته به حالت پایانی می‌رود و توقف می‌کند.

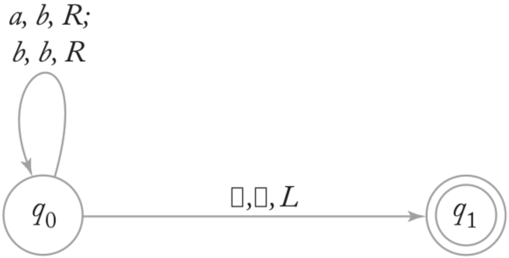
ماشین تورینگ استاندارد

- این ماشین به ازای هر نماد a خوانده شده، آن را به b تغییر می‌دهد و در نهایت با خواندن نماد نانوشته به حالت پایانی می‌رود و توقف می‌کند.
- $\delta(q_0, a) = (q_0, b, R)$ ،$\delta(q_0, b) = (q_0, b, R)$ ،$\delta(q_0, \Box) = (q_1, \Box, L)$



ماشین تورینگ استاندارد

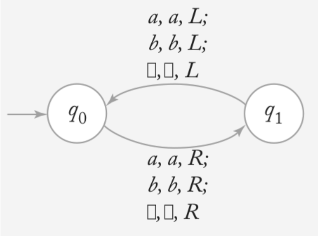
- همانند قبل، ماشین تورینگ را می‌توانیم توسط یک گراف گذار نشان دهیم. بر روی یال‌ها به ترتیب، نماد خوانده شده از روی نوار، نماد نوشته شده بر روی نوار، و جهت حرکت هد درج شده است.
- $\delta(q_0, a) = (q_0, b, R)$ ،$\delta(q_0, b) = (q_0, b, R)$ ،$\delta(q_0, \square) = (q_1, \square, L)$



ماشین تورینگ استاندارد

- یک ماشین تورینگ ممکن است هیچ‌گاه متوقف نشود. در این صورت می‌گوییم ماشین در یک حلقهٔ بی‌پایان افتاده است.[1]

- ماشین تورینگ زیر در زمان اجرا در یک حلقهٔ بی‌پایان می‌افتد.



[1] infinite loop

ماشین تورینگ استاندارد

- ماشین تورینگ را می‌توان به چندین طریق تعریف کرد. یک ماشین تورینگ استاندارد [1] را به صورت زیر تعریف می‌کنیم.

۱. یک ماشین تورینگ نواری دارد که از دو طرف نامحدود است و می‌تواند به طور نامحدود به سمت چپ و راست حرکت کند.

۲. ماشین تورینگ قطعی است، بدین معنی که در هر حالت برای یک نماد در تابع گذار فقط یک حرکت تعریف شده است.

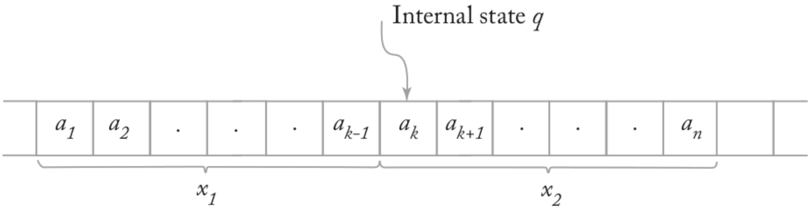
۳. ماشین فایل ورودی و خروجی جداگانه‌ای ندارد. فرض می‌کنیم که قبل از آغاز به کار ماشین، ورودی بر روی نوار نوشته شده باشد. همچنین بعد از توقف ماشین، خروجی بر روی نوار نوشته شده است.

- در آینده با انواع دیگر ماشین تورینگ آشنا می‌شویم.

[1] standard Turing machine

ماشین تورینگ استاندارد

- همانند ماشین‌های پشته‌ای، در ماشین تورینگ از توصیف لحظه‌ای [1] استفاده می‌کنیم.
- هر پیکربندی با توجه به حالت داخلی فعلی ماشین، محتوای نوار، و موقعیت هد ماشین تعیین می‌شود.
- توصیف لحظه‌ای ماشین را با x_1qx_2 یا $a_1 \cdots a_{k-1}qa_k \cdots a_n$ نشان می‌دهیم که بدین معناست که ماشین در حالت q قرار دارد، محتوای نوار $x_1x_2 = a_1 \cdots a_n$ است، و هد ماشین بر روی اولین نماد x_2 یعنی a_k قرار دارد.



[1] instantaneous description

ماشین تورینگ استاندارد

- فرض می‌کنیم که محتوای نوار قبل از x_1 و بعد از x_2 را نمادهای نانوشته تشکیل داده‌اند.
- در صورتی که نماد نانوشته در میانهٔ رشتهٔ ورودی با اهمیت بود آن را نشان می‌دهیم، برای مثال $q\square w$.

ماشین تورینگ استاندارد

- یک حرکت از یک پیکربندی به یک پیکربندی دیگر را با علامت $\vdash$ نشان می‌دهیم.
- بنابراین اگر داشته باشیم $\delta(q_۱, c) = (q_۲, e, R)$ آنگاه حرکت $abq_۱cd \vdash abeq_۲d$ انجام می‌شود، در صورتی که محتوای نوار $abcd$ باشد و ماشین در حالت $q_۱$ قرار داشته باشد و هد ماشین بر روی حرف c باشد.
- نماد $\overset{*}{\vdash}$ برای حرکت در چند گام نشان داده می‌شود.
- همچنین می‌نویسیم $\vdash_M$ اگر حرکت برای ماشین M را در نظر داشته باشیم.

ماشین تورینگ استاندارد

- فرض کنید $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ یک ماشین تورینگ باشد.
- آنگاه رشتهٔ $a_1 \cdots a_{k-1} q_1 a_k \cdots a_n$ یک توصیف لحظه‌ای از ماشین M است به طوری که $a_i \in \Gamma$ و $q_1 \in Q$.
- حرکت $a_1 \cdots a_{k-1} q_1 a_k a_{k+1} \cdots a_n \vdash a_1 \cdots a_{k-1} b q_2 a_{k+1} \cdots a_n$ امکان پذیر است اگر و تنها اگر داشته باشیم: $\delta(q_1, a_k) = (q_2, b, R)$.
- حرکت $a_1 \cdots a_{k-1} q_1 a_k a_{k+1} \cdots a_n \vdash a_1 \cdots q_2 a_{k-1} b a_{k+1} \cdots a_n$ امکان پذیر است اگر و تنها اگر داشته باشیم: $\delta(q_1, a_k) = (q_2, b, L)$.
- با شروع از پیکربندی $x_1 q_i x_2$ ماشین متوقف می‌شود اگر در یک یا چند گام به پیکربندی $y_1 q_j a y_2$ برود $x_1 q_i x_2 \overset{*}{\vdash} y_1 q_j a y_2$ به طوری که $\delta(q_j, a)$ تعریف نشده باشد.
- محاسبه [1] دنباله‌ای از پیکربندی‌های ماشین است که به توقف می‌انجامد.

[1] computation

ماشین تورینگ استاندارد

- اگر یک ماشین با شروع از پیکربندی x_1qx_2 هیچگاه متوقف نشود و در یک حلقهٔ بی‌پایان بیافتد می‌نویسیم $x_1qx_2 \overset{*}{\vdash} \infty$.

ماشین تورینگ استاندارد

- ماشین تورینگ می‌تواند به عنوان یک پذیرنده نیز در نظر گرفته شود.
- اگر رشتهٔ w روی نوار ماشین نوشته شود و بقیهٔ نوار را نمادهای نانوشته تشکیل دهند، ماشین می‌تواند در حالت آغازی با هدی بر روی اولین نماد رشتهٔ w آغاز به کار کند. در صورتی که بعد از تعدادی حرکت، ماشین به یک حالت پایانی رفته و توقف کند، رشته پذیرفته می‌شود.
- فرض کنید $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ یک ماشین تورینگ باشد. زبان پذیرفته شده توسط این زبان برابر است با:
- $L(M) = \{w \in \Sigma^+ : q_0 w \overset{*}{\vdash} x_1 q_f x_2, q_0 \in Q, q_f \in F, x_1, x_2 \in \Gamma^*\}$
- رشته پذیرفته نمی‌شود اگر ماشین در یک حالت غیر پایانی توقف کند و یا اگر ماشین هیچ گاه توقف نکند و در حلقهٔ بی‌پایان بیافتد.

ماشین تورینگ استاندارد

- پس برای محدود کردن رشتهٔ ورودی آن را با نمادهای نانوشته از چپ و راست محصور می‌کنیم.
- بدین ترتیب می‌توانیم در نوار نامحدود، مکان رشته را مشخص کنیم. در غیر اینصورت ماشین هیچ راهی جز جستجو بر روی نوار نامحدود برای نمادهای ورودی نداشت و هیچ گاه نمی‌توانستیم مشخص کنیم آیا حرکت‌های ماشین پایان می‌پذیرد یا خیر.

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که زبان $L = \{a^n b^n : n \geq ۱\}$ را بپذیرد.

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که زبان $L = \{a^n b^n : n \geq ۱\}$ را بپذیرد.
- ماشین را بدین صورت طراحی می‌کنیم که ابتدا با خواندن نماد a آن را با نماد x جایگزین می‌کنیم و هد را به سمت راست حرکت می‌دهیم تا به اولین نماد b برخورد کنیم. نماد b را با نماد y جایگزین می‌کنیم و هد را به سمت چپ حرکت می‌دهیم تا به نماد اولین نماد a بعد از یک نماد x برخورد کنیم. دوباره نماد a را با نماد x جایگزین می‌کنیم و این کار را آنقدر ادامه می‌دهیم تا همه نمادهای a با x و همهٔ b با y جایگزین شود. اگر هیچ نماد a یا b باقی نماند، تعداد نمادهای a و b مساوی بوده است و رشته باید پذیرفته شود.

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که زبان $L = \{a^n b^n : n \geq ۱\}$ را بپذیرد.
- $Q = \{q_۰, q_۱, q_۲, q_۳, q_۴\}, F = \{q_۴\}, \Sigma = \{a, b\}, \Gamma = \{a, b, x, y, \Box\}$
- ابتدا با حرکت دادن هد به سمت راست، به ازای یک نماد a یک نماد b پیدا می‌کنیم، a را با x و b را با y جایگزین می‌کنیم:

$$\delta(q_۰, a) = (q_۱, x, R)$$
$$\delta(q_۱, a) = (q_۱, a, R)$$
$$\delta(q_۱, y) = (q_۱, y, R)$$
$$\delta(q_۱, b) = (q_۲, y, L)$$

- سپس به حالت $q_۲$ می‌رویم و در این حالت هد را آنقدر به سمت چپ حرکت می‌دهیم تا اولین نماد a را پیدا کنیم:

$$\delta(q_۲, y) = (q_۲, y, L)$$
$$\delta(q_۲, a) = (q_۲, a, L)$$
$$\delta(q_۲, x) = (q_۰, x, R)$$

ماشین تورینگ استاندارد

- در پایان وقتی همهٔ نمادهای a و b جایگزین شدند، ماشین در حالت q_0 با یک نماد y مواجه می‌شود.
بنابراین باید از همهٔ نمادهای y عبور کند تا به نماد نانوشته برسد و رشته را بپذیرد.
$$\delta(q_0, y) = (q_3, y, R)$$
$$\delta(q_3, y) = (q_3, y, R)$$
$$\delta(q_3, \square) = (q_4, \square, R)$$

ماشین تورینگ استاندارد

- در صورتی که رشته‌ای متعلق به این زبان نباشد، چند احتمال ممکن است وجود داشته باشد:

۱. رشتهٔ a^*b^* دریافت شده و تعداد نمادهای a از تعداد b بیشتر است. در اینصورت با حرکت به سمت راست در حالت q_1 ماشین هیچ نماد b به ازای یک نماد a پیدا نمی‌کند و با خواندن نماد نانوشته در حالت q_1 متوقف می‌شود.

۲. رشتهٔ a^*b^* دریافت شده و تعداد نمادهای a از نمادهای b کمتر است و در اینصورت ماشین در حالت q_3 به نماد b برخورد می‌کند و متوقف می‌شود.

۳. رشته‌ای غیر از a^*b^* دریافت شده و ماشین در یکی از حالت‌ها متوقف می‌شود.

ماشین تورینگ استاندارد

- در صورتی که رشتهٔ aabb دریافت شود، دنبالهٔ حرکت‌های ماشین به صورت زیر خواهد بود:

- $$\begin{aligned} q_0aabb &\vdash xq_1abb \vdash xaq_1bb \vdash xq_2ayb \\ &\vdash q_2xayb \vdash xq_0ayb \vdash xxq_1yb \\ &\vdash xxyq_1b \vdash xxq_2yy \vdash xq_2xyy \\ &\vdash xxq_0yy \vdash xxyq_3y \vdash xxyyq_3\square \\ &\vdash xxyy\square q_4\square \end{aligned}$$

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که زبان $L = \{a^n b^n c^n : n \geq 1\}$ را بپذیرد.

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که زبان $L = \{a^n b^n c^n : n \geq ۱\}$ را بپذیرد.
- مشابه مسأله قبل به ازای یک نماد a یک نماد b و یک نماد c پیدا می‌کنیم. نماد a را با x ، نماد b را با y و نماد c را با z جایگزین می‌کنیم و در پایان بررسی می‌کنیم که هیچ نماد a یا b یا c باقی نمانده باشد.
- پس ماشین تورینگ علاوه بر زبان $\{a^n b^n\}$ که مستقل از متن است، زبان $\{a^n b^n c^n\}$ را که مستقل از متن نیست را می‌پذیرد و بنابراین قدرت آن از ماشین پشته‌ای بیشتر است.

ماشین تورینگ استاندارد

- از آنجایی که ماشین تورینگ علاوه بر خواندن ورودی از روی نوار، می‌تواند یک خروجی نیز تولید کند، بنابراین این ماشین نه تنها به عنوان یک پذیرنده، بلکه به عنوان یک مبدل نیز می‌تواند مورد استفاده قرار بگیرد.
- ماشین مبدل تورینگ M تابع f را که با رابطهٔ $\widehat{w} = f(w)$ تعریف شده است، پیاده‌سازی می‌کند، اگر $q_0 w \overset{*}{\vdash}_M q_f \widehat{w}$ به طوری که q_f یک حالت پایانی است.

ماشین تورینگ استاندارد

- تابع f با دامنهٔ D تورینگ–محاسبه‌پذیر [1] یا محاسبه‌پذیر [2] نامیده می‌شود اگر یک ماشین تورینگ $M = (Q, \Sigma, \Gamma, \delta, q_\circ, \Box, F)$ وجود داشته باشد به طوری که

$$q_\circ w \vdash_M^* q_f f(w) \quad , \quad q_f \in F$$

به ازای هر $w \in D$

[1] Turing-computable
[2] computable

ماشین تورینگ استاندارد

- به ازای دو عدد صحیح مثبت x و y داده شده، یک ماشین تورینگ طراحی کنید که $x + y$ را محاسبه کند.

ماشین تورینگ استاندارد

- به ازای دو عدد صحیح مثبت x و y داده شده، یک ماشین تورینگ طراحی کنید که $x + y$ را محاسبه کند.
- ابتدا باید روشی برای نمایش یک عدد صحیح ارائه کنیم که بتوان عمل جمع را با استفاده از آن به سادگی انجام داد. برای این کار از نمایش یگانی [1] استفاده می‌کنیم.
- در نمایش یگانی به ازای عدد صحیح مثبت x داریم $w(x) \in \{۱\}^+$ به طوری که $|w(x)| = x$
- همچنین برای عملگر جمع از نماد صفر استفاده می‌کنیم و دو عدد را با یک نماد صفر از یکدیگر جدا می‌کنیم.
- در نهایت نتیجهٔ جمع دو عدد را با یک نماد صفر در پایان بر روی نوار می‌نویسیم.
- پس داریم $q_{\circ} w(x) \circ w(y) \overset{*}{\vdash} q_f w(x+y) \circ$

[1] unary notation

ماشین تورینگ استاندارد

- پس ماشین تورینگ را طوری طراحی می‌کنیم که نماد صفر بین دو عدد را به یک تبدیل کند و آخرین نماد یک در عدد y را به صفر تبدیل کند.

- $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F), Q = \{q_0, q_1, q_2, q_3, q_4\}, F = \{q_4\}$

- $$\begin{aligned} \delta(q_0, 1) &= (q_0, 1, R) \\ \delta(q_0, 0) &= (q_1, 1, R) \\ \delta(q_1, 1) &= (q_1, 1, R) \\ \delta(q_1, \square) &= (q_2, \square, L) \\ \delta(q_2, 1) &= (q_3, 0, L) \\ \delta(q_3, 1) &= (q_3, 1, L) \\ \delta(q_3, \square) &= (q_4, \square, R) \end{aligned}$$

ماشین تورینگ استاندارد

- بنابراین برای جمع دو عدد ۳ و ۲ حرکت‌های زیر را در ماشین داریم:

$$
\begin{aligned}
& q_۰ ۱۱۱۰۱۱ \vdash ۱q_۰ ۱۱۰۱۱ \vdash ۱۱q_۰ ۱۰۱۱ \vdash ۱۱۱q_۰ ۰۱۱ \\
& \vdash ۱۱۱۱q_۱ ۱۱ \vdash ۱۱۱۱۱q_۱ ۱ \vdash ۱۱۱۱۱۱q_۱ \square \\
& \vdash ۱۱۱۱۱q_۲ ۱ \vdash ۱۱۱۱q_۳ ۱۰ \\
& \overset{*}{\vdash} q_۳ \square ۱۱۱۱۱۰ \vdash q_۴ ۱۱۱۱۱۰
\end{aligned}
$$

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که به ازای دو عدد x و y داده شده، در یک حالت پایانی q_y توقف کند اگر $y \geq x$ و در یک حالت غیرپایانی q_n توقف کند، اگر $y < x$.

- به عبارت دیگر

اگر $y \geq x$ آنگاه $q_\circ w(x) \circ w(y) \overset{*}{\vdash} q_y w(x) \circ w(y)$

اگر $y < x$ آنگاه $q_\circ w(x) \circ w(y) \overset{*}{\vdash} q_n w(x) \circ w(y)$

ماشین تورینگ استاندارد

- یک ماشین تورینگ طراحی کنید که به ازای دو عدد x و y داده شده، در یک حالت پایانی q_y توقف کند اگر $x \geq y$ و در یک حالت غیرپایانی q_n توقف کند، اگر $x < y$

- همانند ماشینی که برای زبان $a^n b^n$ طراحی کردیم، این بار ماشینی برای زبان $1^m 0 1^n$ طراحی می‌کنیم. در پایان، اگر تعداد یک باقیمانده قبل از صفر بیشتر بود عدد x بزرگتر است و در غیر اینصورت عدد y بزرگتر است.

- پس در صورتی که $x > y$ بر روی نوار خواهیم داشت: $xx\cdots 11 0 xx \cdots x\square$
و در صورتی که $x < y$ بر روی نوار خواهیم داشت: $xx\cdots xx 0 xx \cdots x 11\square$

- اگر x بزرگتر باشد، به ازای یک نماد یک در قسمت اول رشته، نماد یک در قسمت دوم پیدا نمی‌کنیم و به حالت q_y می‌رویم.

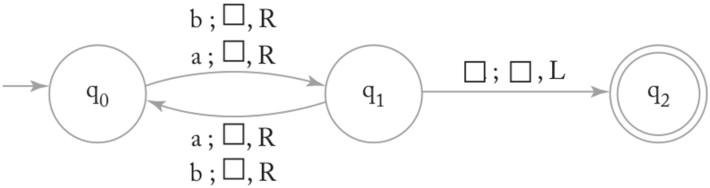
- اگر y بزرگتر باشد، وقتی همهٔ نمادهای یک قسمت اول تبدیل شدند حداقل یک نماد یک در قسمت دوم پیدا می‌کنیم و به حالت q_n می‌رویم.

مثال

- ماشین تورینگی طراحی کنید که زبان $L = \{w : |w| \text{ فرد است}\}$ را بپذیرد.

مثال

- ماشین تورینگی طراحی کنید که زبان $L = \{w : \ |w| \ \text{فرد است}\}$ را بپذیرد.

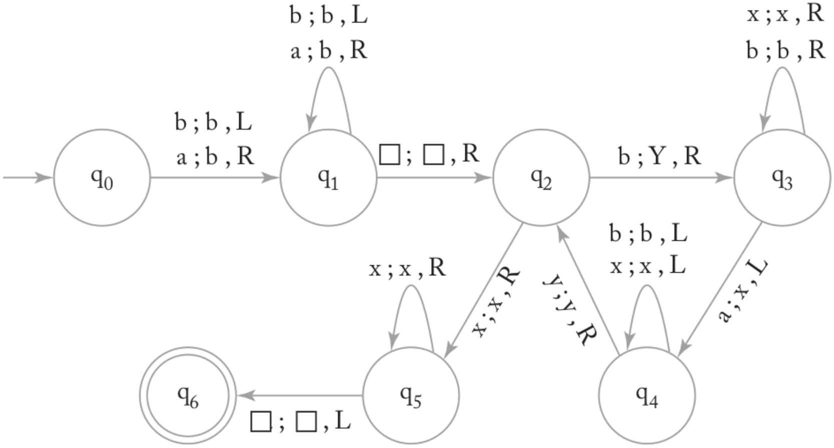


مثال

- ماشین تورینگی طراحی کنید که زبان $L = \{a^n b^m a^{n+m} : n \geq ۰, m \geq ۱\}$ را بپذیرد.

مثال

- ماشین تورینگی طراحی کنید که زبان $L = \{a^n b^m a^{n+m} : n \geq ۰, m \geq ۱\}$ را بپذیرد.

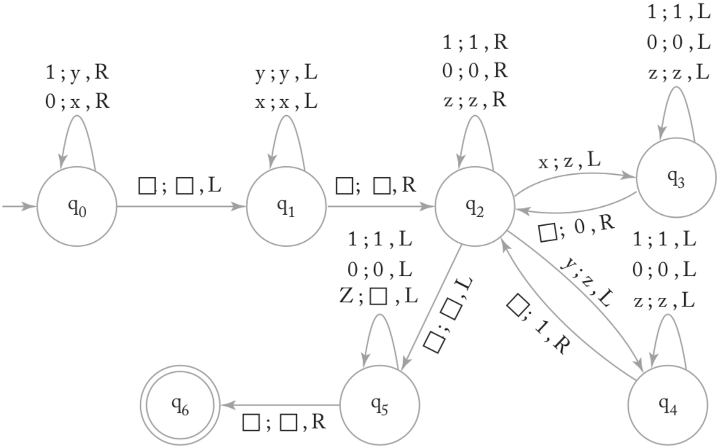


مثال

- ماشین تورینگی طراحی کنید که تابع $f(w) = w^R$ را به ازای $w \in \{۰, ۱\}^+$ محاسبه کند.

مثال

- ماشین تورینگی طراحی کنید که تابع $f(w) = w^R$ را به ازای $w \in \{۰, ۱\}^+$ محاسبه کند.



ترکیب ماشین‌های تورینگ

- دیدیم چگونه می‌توانیم عملیات ساده‌ای را توسط ماشین تورینگ انجام دهیم.
- برای عملیات پیچیده‌تر می‌توانیم این عملیات ساده را با یکدیگر ترکیب کنیم.

ترکیب ماشین‌های تورینگ

- یک ماشین تورینگ طراحی کنید که تابع زیر را محاسبه کند:

  اگر $x \geq y$ آنگاه $f(x, y) = x + y$

  اگر $x < y$ آنگاه $f(x, y) = ۰$

ترکیب ماشین‌های تورینگ

- یک ماشین تورینگ طراحی کنید که تابع زیر را محاسبه کند:
  اگر $x \geq y$ آنگاه $f(x, y) = x + y$
  اگر $x < y$ آنگاه $f(x, y) = ۰$

- از آنجایی که ماشین تورینگ را برای مقایسه و جمع طراحی کرده‌ایم از این پس به جای آن ماشین‌ها می‌توانیم از توصیف سطح بالا استفاده کنیم و آنها را توسط نام عملیاتشان نشان دهیم.

- بعد از اینکه دو عدد را مقایسه کردیم به حالتی می‌رویم که آن حالت، حالت آغازی برای ماشین تورینگی است که عملیات بعدی را انجام می‌دهد.

- پس می‌توانیم دو عدد را توسط ماشین تورینگ مقایسه‌گر C مقایسه کنیم و سپس اگر عدد اول بزرگتر بود یا دو عدد مساوی بودند، عملیات ماشین تورینگ جمع‌کنندهٔ A را برای محاسبهٔ مجموع آغاز می‌کنیم و اگر عدد دوم بزرگتر بود عملیات ماشین تورینگ صفرکننده E را برای تولید خروجی صفر آغاز می‌کنیم.

ترکیب ماشین‌های تورینگ

- پس بعد از مقایسهٔ دو عدد توسط ماشین تورینگ C ، اگر عدد x از عدد y بزرگتر بود، به حالت $q_{A,0}$ می‌رویم تا دو عدد را با یکدیگر جمع کنیم و در غیر اینصورت به حالت $q_{E,0}$ تا اعداد روی نوار را پاک کرده و عدد صفر را بر روی نوار بنویسیم.

- اگر $x \geq y$ آنگاه $q_{C,0} w(x) 0 w(y) \overset{*}{\vdash} q_{A,0} w(x) 0 w(y)$

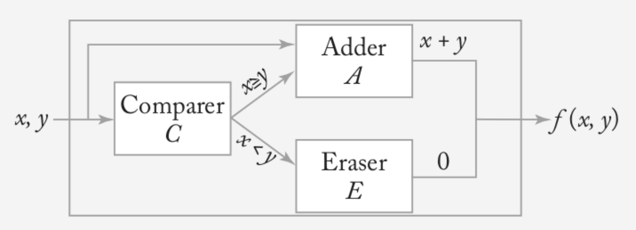
- اگر $x < y$ آنگاه $q_{C,0} w(x) 0 w(y) \overset{*}{\vdash} q_{E,0} w(x) 0 w(y)$

- $q_{A,0} w(x) 0 w(y) \overset{*}{\vdash} q_{A,f} w(x+y) 0$

- $q_{E,0} w(x) 0 w(y) \overset{*}{\vdash} q_{E,f} 0$

ترکیب ماشین‌های تورینگ

- یک ماشین تورینگ طراحی کنید که تابع زیر را محاسبه کند:
  اگر $x \geq y$ آنگاه $f(x, y) = x + y$
  اگر $x < y$ آنگاه $f(x, y) = ۰$



ترکیب ماشین‌های تورینگ

- برای توصیف سطح بالای یک ماشین تورینگ همچنین می‌توانیم از شبه‌کد [1] استفاده کنیم.
- شبه‌کد روشی است برای توصیف عملکرد یک ماشین توسط یک زبان سطح بالا نزدیک به زبان طبیعی انسان.
- گرچه این شبه‌کدها توسط کامپیوترها قابل فهم نیستند اما فرض می‌کنیم که روشی برای تبدیل آنها به زبان ماشین وجود دارد.

[1] pseudocode

ترکیب ماشین‌های تورینگ

- همچنین می‌توانیم از مفهوم زیربرنامه استفاده کنیم، بدین معنی که ماشین اول (یا به طور دقیق‌تر زیرماشین اول) ماشین دوم را برای اجرا فراخوانی می‌کند. ماشین دوم مقدار مورد نیاز ماشین اول را محاسبه می‌کند و نتیجه را روی نوار می‌نویسد. ماشین اول مجددا محاسبات خود را از سر می‌گیرد.

ترکیب ماشین‌های تورینگ

- یک ماشین تورینگ طراحی کنید که دو عدد را در هم ضرب می‌کند.

ترکیب ماشین‌های تورینگ

- یک ماشین تورینگ طراحی کنید که دو عدد را در هم ضرب می‌کند.
- در اینجا از توصیف سطح بالا استفاده می‌کنیم.
- دو عدد x و y را در نمایش یگانی در نظر می‌گیریم. برای ضرب عدد x در عدد y باید به ازای هر نماد یک در عدد x عدد y را روی نوار بنویسیم. سپس همهٔ عددهای y نوشته شده بر روی نوار را با هم جمع کنیم.
- پس به طور دقیق‌تر (۱) به ازای هر نماد یک در عدد x آن را با نماد a جایگزین می‌کنیم و به حالتی می‌رویم که در آن عدد y را روی نوار تکرار می‌کنیم و به حالت اولیه بازمی‌گردیم. (۲) این کار را ادامه می‌دهیم تا هیچ یک از نمادهای یک در عدد x باقی نماند، سپس به حالتی می‌رویم که در آن همهٔ اعداد y نوشته شده بر روی نوار را با هم جمع کنیم. (۳) در پایان برای بازیابی عدد x می‌توانیم همهٔ نمادهای a را با نماد یک جایگزین کنیم.

مثال

- ماشین تورینگی طراحی کنید که یک عدد یگانی را به یک عدد دودویی تبدیل کند. به عبارت دیگر این ماشین باید طول یک رشته شامل نمادهای ۱ را بشمارد و حاصل را در مبنای دو بر روی نوار بنویسد.

مثال

- ماشین تورینگی طراحی کنید که یک عدد یگانی را به یک عدد دودویی تبدیل کند. به عبارت دیگر این ماشین باید طول یک رشته شامل نمادهای ۱ را بشمارد و حاصل را در مبنای دو بر روی نوار بنویسد.
- برای تبدیل یک عدد به معادل آن در مبنای ۲ ، آن عدد را به ۲ تقسیم می‌کنیم و باقیمانده را مرتبهٔ 2^0 می‌نویسیم. سپس خارج قسمت به دست آمده را به ۲ تقسیم می‌کنیم و باقیماندهٔ بعدی را در مرتبهٔ 2^1 می‌نویسیم. این کار را ادامه می‌دهیم تا جایی که خارج قسمت به دست آمده ۰ شود.
- به عبارت دیگر برای به دست آوردن معادل عدد x در مبنای ۲ چنین عمل می‌کنیم:
$$x = 2x_0 + p_0, x_0 = 2x_1 + p_1, x_1 = 2x_2 + p_2, \cdots, x_n = p_n$$
- از بسط دادن عدد x به دست می‌آوریم: $x = 2^n p_n + 4p_2 + 2p_1 + p_0$
- $(x)_{10} = (p_n p_{n-1} \cdots p_2 p_1 p_0)_2$

مثال

- بنابراین ماشین تورینگی که طراحی می‌کنیم، بر روی ورودی ۱۱۱۰۰۰۱۱۱ چنین عمل می‌کند:

۱. در هر گام i (با شروع از $i = 0$) به ازای هر دو نماد یک، یکی را حذف می‌کنیم. برای حذف کردن یک نماد، آن را با علامتی جایگزین می‌کنیم. در این مرحله در واقع عدد را بر دو تقسیم می‌کنیم.

۲. اگر تعداد نمادهای ۱ زوج بود، در پایان تقسیم، باقیماندهٔ تقسیم صفر می‌شود (یعنی به ازای هر نماد ۱ ، یک زوج حذف شونده وجود دارد)، پس $p_i = 0$، بنابراین صفر را در سمت چپ خروجی یادداشت می‌کنیم. اگر تعداد نمادهای ۱ فرد بود، باقیماندهٔ تقسیم ۱ می‌شود، پس یک نماد ۱ باقی می‌ماند که برای آن زوج حذف شونده وجود ندارد، پس $p_i = 1$ و بنابراین عدد ۱ را در سمت چپ خروجی یادداشت می‌کنیم.

۳. به مرحلهٔ ۱ می‌رویم و دوباره نمادهای ۱ را به دو تقسیم می‌کنیم و این کار را ادامه می‌دهیم تا همهٔ نمادهای ورودی حذف شوند و ماشین را در حالت پایانی متوقف می‌کنیم.

مثال

- ماشین تورینگی طراحی کنید که یک عدد دودویی را به یک عدد یگانی تبدیل کند.

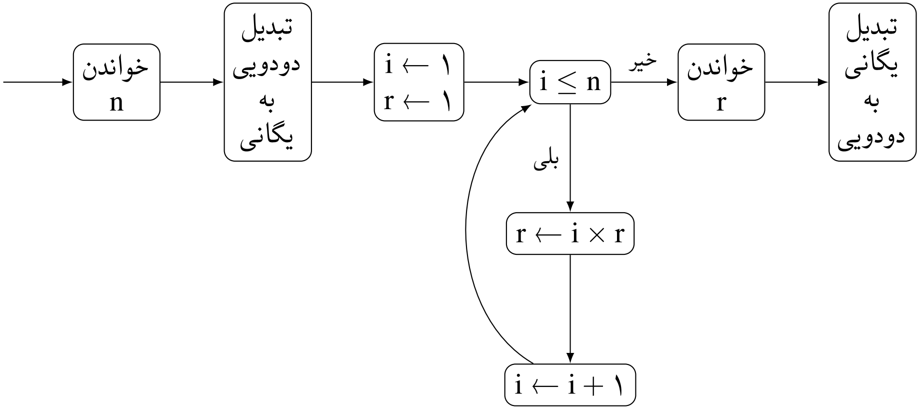
مثال

- ماشین تورینگی طراحی کنید که یک عدد دودویی را به یک عدد یگانی تبدیل کند.
- الگوریتم کلی بدین ترتیب عمل می‌کند: در هر مرحله n امین نماد از رشتهٔ دودویی را از سمت راست می‌خوانیم. در ابتدا قرار می‌دهیم $n = ۱$. مرتبهٔ نماد (بیت) خوانده شده برابر است با $m = ۲^i$. در ابتدا قرار می‌دهیم $i = ۰$. مقدار اولیه m را که برابر با ۱ است در قسمتی از نوار می‌نویسیم. حاصل به دست آمده از تبدیل عدد دودویی برابر است با r . در ابتدا قرار می‌دهیم $r = ۰$.

۱. بیت n ام را می‌خوانیم. مقدار آن را در m ضرب می‌کنیم. حاصل به دست آمده را به مقدار حاصل خروجی r اضافه می‌کنیم.
۲. مقدار n را یک واحد می‌افزاییم (یک سلول به سمت چپ می‌رویم). مقدار m را دو برابر می‌کنیم (تعداد نمادهای یک آن را کپی می‌کنیم).
۳. اگر مقدار بیت n ام برابر با نانوشته نبود به مرحله ۱ می‌رویم، در غیر اینصورت ماشین در حالت پایانی متوقف می‌شود.

مثال

- ماشین تورینگی را توصیف کنید که عدد n فاکتوریل را محاسبه کند.



تز تورینگ

- نشان دادیم که ماشین تورینگ نه تنها برای محاسبات ساده، بلکه برای محاسبات پیچیده‌تر نیز با ترکیب ماشین‌های مختلف می‌تواند مورد استفاده قرار بگیرد.
- تا اینجا متوجه شدیم که ماشین تورینگ از ماشین‌های پشته‌ای قدرتمند تر است چنانچه با یک مثال نشان دادیم که ماشین تورینگ زبانی را می‌پذیرد که ماشین‌های پشته‌ای نمی‌پذیرند.
- همچنین نشان دادیم که عملیات ساده مانند مقایسه، جمع و ضرب، و کپی کردن یک مقدار با استفاده از ماشین تورینگ امکان پذیر است و از آنجایی که عملیات پیچیده‌تر از ترکیب این عملیات مقدماتی به دست می‌آیند، می‌توانیم حدس بزنیم که ماشین تورینگ هر محاسبهٔ پیچیده‌ای را می‌تواند انجام دهد.
- اما آیا ماشین تورینگ همهٔ محاسباتی را که با هر ماشین دیگری قابل انجام است می‌تواند انجام دهد؟

تز تورینگ

- اگر بخواهیم ماشین تورینگ را با یک کامپیوتر دیجیتال مقایسه کنیم، کافی است که دستورات کامپیوتر مورد نظر را یک به یک با ماشین‌های تورینگ متناظرشان مقایسه کنیم. از آنجایی که یک محاسبهٔ پیچیده از ترکیب تعدادی دستور تشکیل شده است که توسط ماشین‌های تورینگ قابل شبیه‌سازی هستند، و از آنجایی که ماشین‌های تورینگ را نیز می‌توانیم با هم ترکیب کنیم، پس قدرت ماشین تورینگ باید به اندازهٔ ماشین دیجیتال مورد نظر باشد.
- اگر بتوانیم محاسباتی پیدا کنیم و نشان دهیم که آن محاسبات توسط ماشین دیگری قابل انجام است، ولی هیچ ماشین تورینگی برای آن وجود ندارد، آنگاه به این نتیجه می‌رسیم که ماشینی قدرتمندتر از ماشین تورینگ وجود دارد.
- اما کسی تاکنون چنین محاسباتی پیدا نکرده است، پس حدس می‌زنیم که ماشین تورینگ می‌تواند هر محاسبه‌ای را که توسط هر ماشین دیگری انجام شود را انجام دهد.

تز تورینگ

- تز تورینگ [1] که در سال ۱۹۳۶ توسط آلن تورینگ [2] بیان شد، می‌گوید که هر محاسبه‌ای که توسط یک محاسبه‌گر مکانیکی انجام شود، می‌تواند توسط ماشین تورینگ نیز انجام شود.
- این فرضیه را نمی‌توان اثبات کرد، زیرا باید به طور دقیق و رسمی بیان کنیم منظور از محاسبه‌گر مکانیکی چیست. قبل از کامپیوترهای دیجیتال که در سال ۱۹۴۵ به وجود آمدند، محاسبه‌گرهای مکانیکی برای محاسبات استفاده می‌شدند. همچنین می‌توان محاسبه بر روی کاغذ توسط انسان را یک محاسبهٔ مکانیکی محسوب کرد.
- اما آیا کامپیوترهایی که در آینده به وجود می‌آیند قدرتمندتر از ماشین تورینگ خواهند بود و آیا محاسباتی یافت خواهد شد که توسط ماشین تورینگ قابل انجام نباشند؟

[1] Turing thesis
[2] Alan Turing

تز تورینگ

- در حال حاضر می‌دانیم که:

۱. هر محاسبه‌ای که توسط یک کامپیوتر دیجیتال قابل انجام است توسط ماشین تورینگ نیز قابل انجام است.

۲. هیچ کس هیچ مسألهٔ محاسباتی پیدا نکرده است که توسط ماشین تورینگ قابل انجام نباشد.

۳. مدل‌های محاسباتی دیگری (مانند حساب لامبدا [1]) ارائه شده‌اند که هیچ کدام از مدل محاسباتی تورینگ قدرتمندتر نیستند و ثابت شده است که همه با یکدیگر هم‌ارزند.

[1] lambda-calculus

تز تورینگ

- با این حال هنوز تز تورینگ در حد فرضیه است.
- می‌توان تز تورینگ را با قوانین فیزیک نیوتون مقایسه کرد. درستی قوانین نیوتن اثبات شدنی نیست. تنها می‌دانیم که همهٔ آزمایش‌ها و مشاهدات درستی آنها را تأیید می‌کنند، اما ممکن است شرایطی وجود داشته باشد که در آن قوانین نیوتن نقض شوند. پس این قوانین اثبات نمی‌شوند، و تنها ممکن است با مشاهداتی نقض شوند.
- پس تز تورینگ می‌تواند به منزلهٔ قوانین پایه‌ای علوم کامپیوتر در نظر گرفته شود.

تز تورینگ

- یک الگوریتم [1] برای تابع $f : D \to R$ یک ماشین تورینگ M است، که به ازای دریافت ورودی $d \in D$ خوانده شده از روی نوار خود، در نهایت با جواب $f(d) \in R$ نوشته شده بر روی نوار خود، متوقف می‌شود.
می‌نویسیم $q_f \in F$, $q_\circ d \overset{*}{\vdash}_M q_f f(d)$ به ازای همهٔ مقادیر $d \in D$

- از این پس می‌توانیم از یک شبه‌کد یا یک زبان سطح بالا برای توصیف محاسبات استفاده کنیم با این اطمینان که برای آن یک ماشین تورینگ وجود دارد.

[1] algorithm

ماشین‌های تورینگ دیگر

- حال به بررسی ماشین‌های تورینگ دیگر می‌پردازیم که گرچه امکانات بیشتری (مانند حافظهٔ بیشتر) دارند ولی قدرت آنها از ماشین تورینگ بیشتر نیست.
- در این بخش به بررسی ماشین‌های تورینگ با تعداد بیشتری نوار، نوار در ابعادی بیشتر از یک بعد، ماشین تورینگ غیرقطعی [1]، و ماشین تورینگ جهانی [2] می‌پردازیم.
- در پایان ماشین‌های کراندار خطی [3] را بررسی می‌کنیم که یک ماشین تورینگ محدود شده است و برای شناسایی زبان‌های حساس به متن [4] به کار می‌رود.

[1] nondeterministic Turing machine
[2] universal Turing machine
[3] linear bounded automata
[4] context-sensitive languages

ماشین‌های تورینگ دیگر

- دو ماشین هم‌ارز [1] یکدیگرند اگر هر دو یک زبان را شناسایی کنند.
- یک دسته (طبقه یا کلاس [2]) از ماشین‌ها، ماشین‌هایی هستند که همگی یک تعریف یکسان دارند. برای مثال هر ماشین متناهی قطعی متعلق به دستهٔ ماشین‌های متناهی قطعی است. تاکنون با سه دسته از ماشین‌ها آشنا شدیم: ماشین‌های متناهی، ماشین‌های پشته‌ای، و ماشین‌های تورینگ. هر کدام از این دسته‌ها دو زیر دستهٔ قطعی و غیرقطعی نیز دارند.

[1] equivalent
[2] class

ماشین‌های تورینگ دیگر

- حال دو دستهٔ C_1 و C_2 از ماشین‌ها را در نظر بگیرید.
- اگر برای هر ماشین M_1 در دستهٔ C_1 یک ماشین M_2 در دستهٔ C_2 وجود داشته باشد به طوری که $L(M_1) = L(M_2)$ باشد، آنگاه می‌گوییم کلاس C_2 قدرتی حداقل برابر با کلاس C_1 دارد.
- اگر همچنین برای هر ماشین M_2 در دستهٔ C_2 یک ماشین M_1 در دستهٔ C_1 وجود داشته باشد به طوری که $L(M_1) = L(M_2)$ باشد، آنگاه می‌گوییم کلاس C_1 و C_2 هم‌ارزند.

ماشین‌های تورینگ با انتخاب توقف

- در ماشین تورینگ استاندارد، همیشه هد به چپ یا راست حرکت می‌کند. گاهی نیاز به توقف نیز می‌باشد که می‌توان گزینهٔ توقف را به ماشین تورینگ افزود. ماشین تورینگ با انتخاب توقف [1] این امکان را به ماشین تورینگ می‌افزاید.
$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$
- در اینجا S به معنی توقف هد در یک سلول است.
- این ماشین هم‌ارز ماشین تورینگ استاندارد است. ایدهٔ اثبات: توقف را می‌توان با یک حرکت به چپ و یک حرکت به راست شبیه‌سازی کرد.

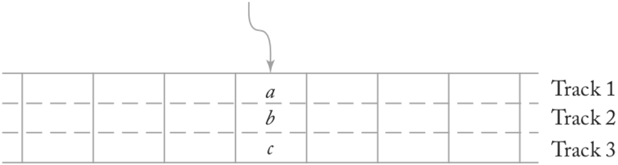
[1] Turing machine with a stay-option

ماشین‌های تورینگ با چند شیار

- در تعریف ماشین تورینگ گاهی به جای نوشتن یک نماد در یک سلول، یک رشته در یک سلول می‌نویسند. می‌توان نشان داد که این تعریف با تعریف ماشین تورینگ استاندارد یکسان است.

- حال اگر به جای نوشتن یک نماد در یک سلول، یک سه‌تایی در یک سلول بنویسیم، همانند این است که یک سلول را شیاربندی کرده‌ایم.

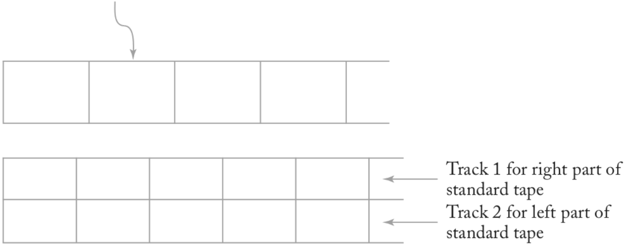
- چنین ماشینی را ماشین تورینگ با چند شیار [1] می‌نامیم که هم‌ارز ماشین تورینگ استاندارد است.



[1] Turing machine with multiple track

ماشین‌های تورینگ با نوار نیمه‌نامحدود

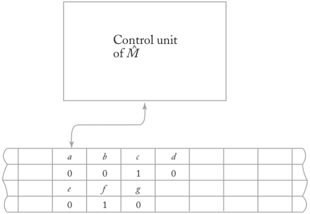
– اگر یک ماشین تورینگ با نوار نیمه‌نامحدود [1] داشته باشیم، می‌توانیم یک ماشین معادل با نوار نیمه‌نامحدود با دو شیار در نظر بگیریم. سپس فرض می‌کنیم هر گاه ماشین از کران سمت چپ نوار به سمت راست حرکت می‌کند، فقط در شیار بالایی می‌نویسد، و هر گاه به کران سمت چپ نوار رسید و با حرکت به چپ از کران گذر کرد در شیار پایینی می‌نویسد. بدین ترتیب این ماشین تورینگ ماشین تورینگ استاندارد را شبیه‌سازی می‌کند.

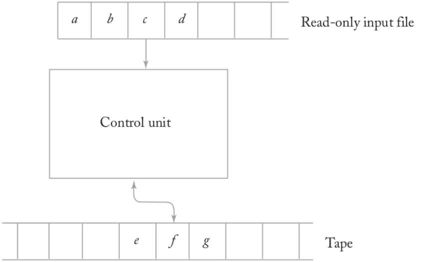


[1] Turing machine with semi-infinite tape

ماشین تورینگ آفلاین

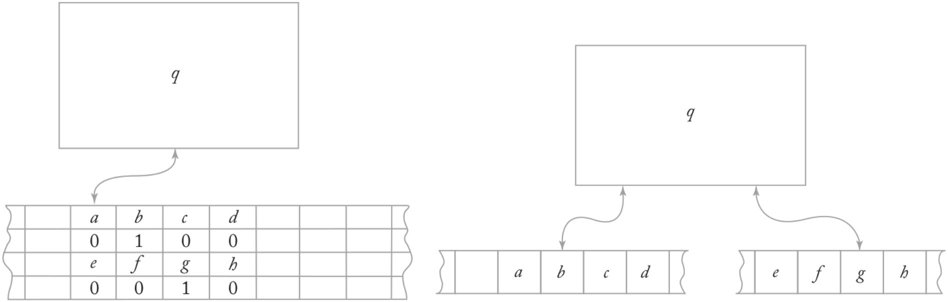
- یک ماشین تورینگ آفلاین ماشینی است که در آن ورودی از هد خواندن و نوشتن جدا شده است. برای شبیه‌سازی این ماشین توسط یک ماشین تورینگ استاندارد از یک ماشین تورینگ با یک نوار چهار شیاری استفاده می‌کنیم. شیار اول ورودی را در بر می‌گیرد، شیار دوم مکان هد خواندن از ورودی، شیار سوم محتوای نوار، و شیار چهارم مکان هد خواندن نوشتن بر روی نوار را در بر می‌گیرد. پس ماشین تورینگ آفلاین را می‌توان توسط یک ماشین تورینگ استاندارد (با نوار چهار شیاری) شبیه‌سازی کرد. بدینگونه می‌توان نشان داد که این ماشین نیز هم‌ارز ماشین تورینگ استاندارد است.





ماشین‌های تورینگ با چند نوار

- یک ماشین تورینگ با چند نوار [1] و به طور مشخص با n نوار را می‌توان توسط یک ماشین تورینگ با یک نوار ۲n شیاری شبیه‌سازی کرد. به طوری که شیار n محتوای نوار n ام، و شیار ۲n موقعیت هد در شیار n ام را نشان می‌دهد.



[1] multitape Turing machine

ماشین‌های تورینگ با نوار چندبعدی

- یک ماشین تورینگ با یک نوار چندبعدی [1] را می‌توان توسط یک ماشین تورینگ با یک نوار دوشیاری شبیه‌سازی کرد به طوری که شیار اول محتوای نوار چندبعدی و شیار دوم مکان محتوا را در نوار چندبعدی مشخص می‌کند.
- به روشی دیگر می‌توان یک ماشین تورینگ با نوار n بعدی را با یک ماشین تویگ با یک نوار شامل $n + ۱$ شیار شبیه‌سازی کرد، به طوری که شیار اول محتوای نوار n بعدی و n شیار دیگر هر کدام مختصات یکی از ابعاد را در برمی‌گیرد.

[1] multidimentional Turing machine

ماشین‌های تورینگ غیرقطعی

- یک ماشین تورینگ غیرقطعی ماشینی همانند ماشین تورینگ قطعی است، با این تفاوت که تابع گذار آن به صورت زیر تعریف می‌شود: $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$
- این بدین معنی است که در هر حرکت، ماشین می‌تواند یکی از گذارهای ممکن را انتخاب کند.
- یک ماشین تورینگ غیرقطعی رشتهٔ w را می‌پذیرد اگر دنباله‌ای از حرکت‌ها وجود داشته باشد که ماشین را با شروع از حالت آغازی و خواندن رشتهٔ w به یک حالت پایانی ببرد.
- با خواندن رشتهٔ w ممکن است دنباله‌های دیگری از حرکت‌ها وجود داشته باشند که ماشین را به حلقهٔ بی‌پایان یا حالت غیرپایانی ببرند.

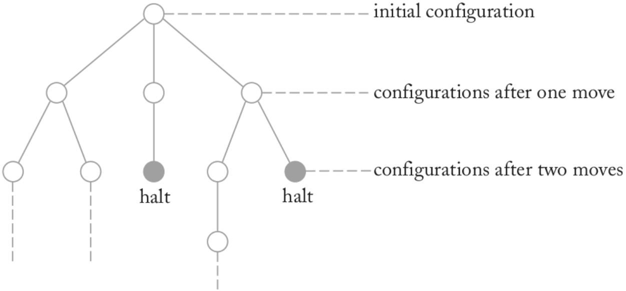
ماشین‌های تورینگ غیرقطعی

- برای اینکه نشان دهیم قدرت ماشین تورینگ غیرقطعی به اندازهٔ قدرت ماشین تورینگ قطعی است باید بتوانیم برای هر ماشین غیرقطعی یک ماشین قطعی ارائه کنیم.

ماشین‌های تورینگ غیرقطعی

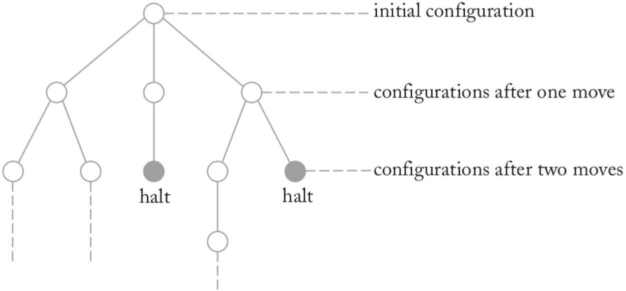
- فرض کنید می‌خواهیم یک الگوریتم (که در واقع یک ماشین تورینگ استاندارد است) را توصیف کنیم که یک ماشین تورینگ غیرقطعی را اجرا می‌کند.

- در هر حرکت از ماشین غیرقطعی (که در آن چندین انتخاب وجود دارد)، الگوریتم مورد نظر، به ازای هر انتخاب، ماشین تورینگ غیرقطعی را در قسمتی از نوار کپی می‌کند و هر کپی از ماشین را جداگانه اجرا می‌کند. اگر یکی از کپی‌ها به حالت پایانی رسید و توقف کرد، الگوریتم مورد نظر رشته ورودی را می‌پذیرد.



ماشین‌های تورینگ غیرقطعی

- اگر تعداد انتخاب‌های یک ماشین در هر حرکت حداکثر k باشد، آنگاه حداکثر تعداد پیکربندی های ایجاد شده بعد از n حرکت برابر است با k^n .



ماشین‌های تورینگ غیرقطعی

- حال که برای اجرای یک ماشین تورینگ غیرقطعی الگوریتمی ارائه کردیم، همین الگوریتم را می‌توانیم به صورت یک ماشین تورینگ قطعی ارائه کنیم. این ماشین تورینگ قطعی معادل ماشین تورینگ غیرقطعی مورد نظر است.
- ماشین تورینگ قطعی طراحی شده، در هر حرکت توصیف لحظه‌ای ماشین غیرقطعی را روی نوار خود می‌نویسد. سپس حرکت‌های مختلف را به ازای انتخاب‌های ماشین غیرقطعی محاسبه می‌کند و این روند را ادامه می‌دهد تا به یک حالت پایانی برسد.
- پس این ماشین تورینگ قطعی، معادل ماشین تورینگ غیرقطعی است و بنابراین به ازای هر ماشین غیرقطعی یک ماشین قطعی وجود دارد و این دو طبقه از ماشین‌ها هم‌ارز یکدیگرند.

ماشین‌های تورینگ غیرقطعی

- یک ماشین تورینگ غیرقطعی زبان L را می‌پذیرد[1]، اگر به ازای هر جملهٔ $w \in L$ یک دنباله از حرکت‌ها وجود داشته باشد (یک پیکربندی برای ماشین پس از چندگام اجرا وجود داشته باشد) که رشتهٔ w را بپذیرد. ممکن است دنباله‌ای از حرکت‌ها در حلقهٔ بی‌پایان بیافتند و دنباله‌ای دیگر بدون پذیرفتن متوقف شود، که چنین رفتاری در پذیرفتن رشته تأثیری ندارد و فقط وجود داشتن دنباله‌ای که به پذیرفتن رشته ختم می‌شود برای پذیرفتن رشته اهمیت دارد.

- یک ماشین تورینگ غیرقطعی می‌تواند زبان L را تصمیم‌بگیرد[2]، (بر روی زبان L تصمیم‌پذیر است) اگر به ازای هر جملهٔ $w \in \Sigma^*$ دنباله‌ای از حرکت‌ها وجود داشته باشد که یا جمله را بپذیرد یا جمله را رد کند. (برای رد کردن جمله، هیچ دنباله‌ای نمی‌تواند به حلقهٔ بی‌پایان برود، زیرا در حلقهٔ بی‌پایان نمی‌توانیم مطمئن باشیم که جمله پذیرفته نخواهد شد.)

[1] accept
[2] decide

ماشین تورینگ جهانی

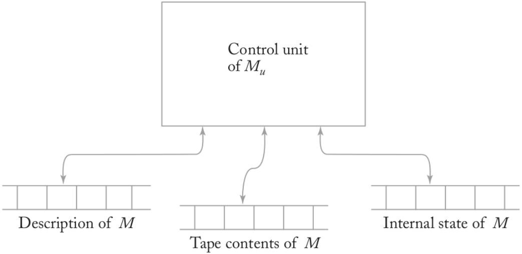
- ماشین تورینگ استاندارد معادل یک الگوریتم است که یک تابع خاص را محاسبه می‌کند.
- بنابراین ماشین تورینگ استاندارد را نمی‌توان به عنوان یک محاسبه‌گر جامع معادل یک کامپیوتر دیجیتال دانست.
- بدین جهت ماشین تورینگ جهانی [1] را تعریف می‌کنیم که یک محاسبه‌گر جامع است و می‌تواند هر محاسبه‌ای را بر روی هر جمله‌ای انجام دهد.
- یک ماشین تورینگ جهانی M_u ماشینی است که با گرفتن توصیف یک ماشین تورینگ M و رشتهٔ w محاسبات M بر روی w را شبیه‌سازی می‌کند.

[1] universal Turing machine

ماشین تورینگ جهانی

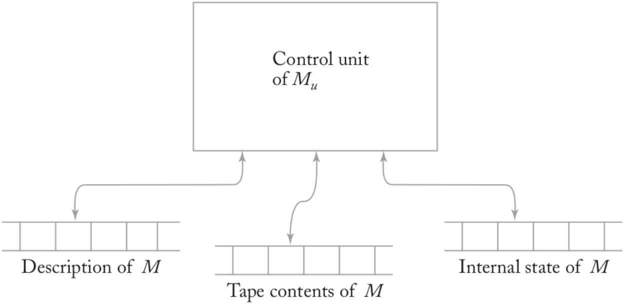
- یک ماشین تورینگ جهانی سه نوار دارد. بر روی نوار اول توصیف یک ماشین تورینگ را دریافت می‌کند، بر روی نوار دوم رشتهٔ ورودی را دریافت می‌کند، و بر روی نوار سوم حالت داخلی ماشین ورودی را ذخیره می‌کند.

- بنابراین با استفاده از نوار دوم و سوم، ماشین تورینگ جهانی توصیف لحظه‌ای ماشین تورینگ دریافتی خود را تولید می‌کند، سپس با استفاده از توابع گذار نوشته شده بر روی نوار اول، ماشین تورینگ جهانی، ماشین تورینگ دریافتی را اجرا می‌کند.



ماشین تورینگ جهانی

- پس همانطور که یک کامپیوتر دیجیتال، یک الگوریتم و مقدار ورودی آن را دریافت می‌کند و خروجی الگوریتم را محاسبه می‌کند، یک ماشین تورینگ جهانی نیز، یک ماشین تورینگ و یک رشتهٔ ورودی را دریافت کرده و محاسبات ماشین تورینگ را بر روی رشتهٔ ورودی محاسبه می‌کند.



ماشین تورینگ جهانی

- ماشین تورینگ ورودی را می‌توان به صورت یک رشته شامل صفر و یک دریافت کرد.
- بدین منظور باید حالت‌های ماشین را با اعداد دودویی و الفبای نوار را نیز با اعداد دودویی کدگذاری کنیم. بنابراین یک تابع گذار برای ماشین تورینگ می‌تواند با یک رشتهٔ دودویی مشخص شود و کل ماشین تورینگ با رشته های دودویی از توابع گذار.
- برای مثال تابع گذار $\delta(q_1, a_2) = (q_2, a_3, L)$ را می‌توان به صورت ۱۰۱۱۰۱۱۰۱۱۱۰۱۰ کدگذاری کرد به طوری که همهٔ صفر ها علامت فاصله هستند، اولین ۱ کدگذاری برای q_1، ۱۱ بعدی کدگذاری برای a_2، ۱۱ بعدی کدگذاری برای q_2، ۱۱۱ بعدی کدگذاری برای a_3 و ۱ نهایی کدگذاری برای L می‌باشد.

ماشین تورینگ جهانی

- از آنجایی که مجموعهٔ همهٔ رشته‌های دودویی قابل شمارش است، پس مجموعهٔ همهٔ ماشین‌های تورینگ نیز قابل شمارش است.
- برای شمارش همهٔ ماشین‌های تورینگ، با شمارش رشته‌های دودویی شروع می‌کنیم. به ازای هر رشتهٔ دودویی در $\{۰, ۱\}^+$، بررسی می‌کنیم آیا رشتهٔ دودویی معادل یک ماشین تورینگ است یا خیر. اگر جواب مثبت بود، ماشین تورینگ مورد نظر را در مجموعهٔ همهٔ ماشین‌های تورینگ شمارش می‌کنیم، در غیر اینصورت به رشتهٔ بعدی می‌رویم.

ماشین‌های کراندار خطی

- اگر ماشین تورینگ را محدود کنیم به طوری که نوار آن مانند یک پشته عمل کند از قدرت آن می‌کاهیم و آن را به یک ماشین پشته‌ای تبدیل می‌کنیم.

- اگر نوار نامحدود را به نوار محدود تبدیل کنیم، یک ماشین متناهی به دست می‌آوریم.

- اگر ماشین را محدود کنیم به طوری که فقط از قسمتی از نوار استفاده کند که رشتهٔ ورودی بر روی آن نوشته شده است، یک ماشین کراندار خطی [1] به دست آوریم.

- ماشین کراندار خطی مانند ماشین تورینگ یک نوار نامحدود دارد، با این تفاوت که فقط از قسمتی از نوار استفاده می‌کند که ورودی بر روی آن قرار دارد.

- بنابراین رشتهٔ ورودی با دو نشانهٔ سمت چپ و سمت راست [2] مشخص می‌شود و ماشین نمی‌تواند از آن محدوده خارج شود.

---

[1] linear bounded automata (lba)

[2] left-end and right-end marker

ماشین‌های کراندار خطی

- ماشین کراندار خطی، یک ماشین تورینگ غیر قطعی $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ است، با این تفاوت که الفبای Σ دو نماد ویژه برای نشانهٔ سمت چپ و راست رشته ($[,]$) دارد.
- همهٔ اعضای $\delta(q_i, [)$ به شکل $(q_j, [, R)$ و همهٔ اعضای $\delta(q_i,])$ به شکل $(q_j,], L)$ هستند.
- رشتهٔ w توسط یک ماشین کراندار خطی پذیرفته می‌شود اگر دنباله‌ای از حرکت‌ها وجود داشته باشد به طوری که $q[w] \overset{*}{\vdash} [x_1 q_f x_2]$ به ازای $q_f \in F$ و $x_1, x_2 \in \Gamma^*$.
- یک زبان پذیرفته شده توسط یک ماشین کراندار خطی مجموعه‌ای از تمام رشته‌های پذیرفته شده توسط آن ماشین است.

ماشین‌های کراندار خطی

- می‌توان نشان داد که قدرت ماشین کراندار خطی بیشتر از ماشین پشته‌ای است زیرا زبان $\{a^n b^n c^n : n \geq ۱\}$ را می‌پذیرد.
- نشان داده شده است که قدرت ماشین کراندار خطی کمتر از ماشین تورینگ است و طبقهٔ زبان‌هایی که می‌پذیرد زیرمجموعهٔ زبان‌هایی است که توسط ماشین‌های تورینگ پذیرفته می‌شوند. در آینده با این دسته از زبان‌ها آشنا می‌شویم.