نظریه‌ی زبان‌ها و ماشین‌ها

درس‌نامه‌ی ۱۰

کاظم فولادی
http://kazim.fouladi.ir

ویراست اول: ۱۳۸۵
ویراست دوم: ۱۳۸۷
ویراست سوم: ۱۳۹۳

فهرست مطالب

۱۰ مدل‌های دیگر ماشین تورینگ

OTHER MODELS OF TURING MACHINES

برای ماشین تورینگ مدل‌های دیگری هم ارائه شده است که همگی به یک اندازه قابل قبول هستند.
قدرت ماشین تورینگ مستقل از ساختار انتخابی برای ماشین است.
اگر تز تورینگ را بپذیریم، باید قبول کنیم که هر چه ساختار حافظه‌ی آن پیچیده‌تر شود، تاثیری در قدرت آن ندارد.
همچنین، قدرت ماشین تورینگ غیرقطعی از ماشین تورینگ قطعی بیشتر نیست.

۱۰-۱ تغییرات جزیی بر روی طرح ماشین تورینگ

ابتدا تغییراتی در تعریف ماشین تورینگ استاندارد اعمال می‌کنیم و اثر آن را در مفهوم کلی ماشین تورینگ در نظر می‌گیریم.

۱۰-۱-۱ هم‌ارزی کلاس‌های آتوماتا

تعریف دو آتوماتون هم‌ارز هستند، اگر زبان یکسانی را بپذیرند. دو کلاس آتوماتا C_1 و C_2 را در نظر می‌گیریم. اگر برای هر آتوماتون M_1 در C_1، آتوماتون M_2 در C_2 وجود داشته باشد به گونه‌ای که

$$L(M_1) = L(M_2)$$

می‌گوییم که C_2 حداقل به اندازه‌ی C_1 قدرتمند *(powerful)* است. اگر عکس این مطلب نیز برقرار باشد و برای هر M_2 در C_2، آتوماتون M_1 در C_1 وجود داشته باشد که $L(M_1) = L(M_2)$، می‌گوییم که C_1 و C_2 هم‌ارز هستند.

شبیه‌سازی. روش‌های متعددی برای برقراری هم‌ارزی بین آتوماتا وجود دارد.
برای نشان دادن هم‌ارزی در بحث ماشین تورینگ، از تکنیک مهم **شبیه‌سازی** استفاده می‌کنیم.
فرض می‌کنیم M یک آتوماتون باشد. می‌گوییم آتوماتون دیگر $\hat{M}$ می‌تواند یک محاسبه از M را شبیه‌سازی کند، اگر $\hat{M}$ بتواند محاسبه‌ی M را به صورت زیر تقلید نماید: اگر $d_0, d_1, d_2, \ldots$ دنباله‌ی توصیفات بلافصل محاسبه‌ی M باشد، یعنی

$$d_0 \vdash_M d_1 \vdash_M \ldots \vdash_M d_n \vdash_M \ldots$$

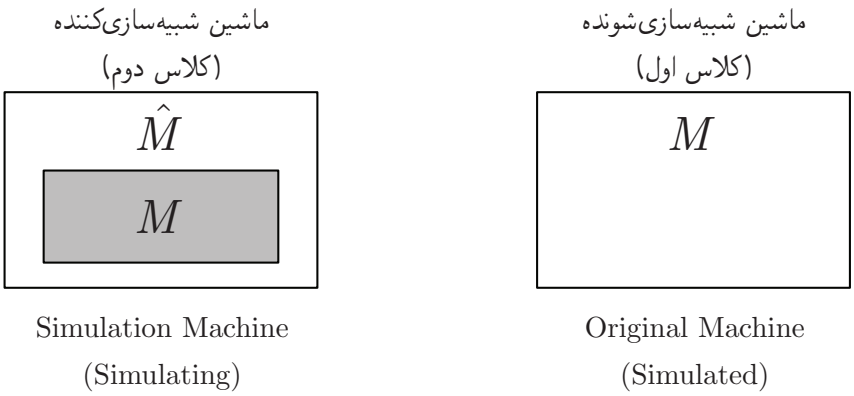
آن‌گاه $\hat{M}$ این محاسبه را شبیه‌سازی می‌کند، اگر محاسبه‌ای قابل قیاس با محاسبه‌ی M انجام دهد:

$$\hat{d}_0 \vdash^*_{\hat{M}} \hat{d}_1 \vdash^*_{\hat{M}} \dots \vdash^*_{\hat{M}} \hat{d}_n \vdash^*_{\hat{M}} \dots$$

که در آن $\hat{d}_0, \hat{d}_1, \hat{d}_2, \dots$ دنباله‌ی توصیفات بلافصل $\hat{M}$ است و هر یک از آنها متناظر با یک پیکربندی یکتا از M باشد.

$$\begin{array}{ccccccccccc} d_0 & \vdash_M & d_1 & \vdash_M & \dots & \vdash_M & d_n & \vdash_M & \dots & \vdash_M & d_f \\ \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow & & \updownarrow \\ \hat{d}_0 & \vdash^*_{\hat{M}} & \hat{d}_1 & \vdash^*_{\hat{M}} & \dots & \vdash^*_{\hat{M}} & \hat{d}_n & \vdash^*_{\hat{M}} & \dots & \vdash^*_{\hat{M}} & \hat{d}_f \end{array}$$

به عبارت دیگر، اگر محاسبات انجام شده با $\hat{M}$ را بدانیم، بتوانیم به طور دقیق محاسبات انجام شده توسط M را تعیین نماییم.
توجه کنید که شبیه‌سازی یک حرکت واحد $d_i \vdash_M d_{i+1}$ از M می‌تواند شامل حرکات متعددی از $\hat{M}$ باشد. پیکربندیهای میانی در $\hat{d}_i \vdash^*_{\hat{M}} \hat{d}_{i+1}$ ممکن است با هیچ پیکربندی در M متناظر نباشد.
اگر $\hat{M}$ بتواند هر محاسبه‌ی M را شبیه‌سازی کند، می‌گوییم $\hat{M}$، M را شبیه‌سازی می‌کند. حال اگر M هم بتواند $\hat{M}$ شبیه‌سازی کند، نتیجه می‌گیریم که هر دو یک زبان را می‌پذیرند و با یکدیگر هم‌ارز می‌باشند.



نتیجه برای نشان دادن هم‌ارزی دو کلاس از آتوماتا، نشان می‌دهیم که برای هر ماشین در یک کلاس، ماشینی در کلاس دیگر وجود دارد که آن را شبیه‌سازی می‌کند، و برعکس.

۲-۱-۱۰ ماشین تورینگ با گزینه‌ی ایست

در تعریف ماشین تورینگ استاندارد، هد خواندن/نوشتن در هر حرکت باید یکی از حرکات به سمت چپ یا راست را انجام دهد. گاهی یک گزینه‌ی دیگر هم مفید است که هد پس از بازنویسی محتوای یک سلول حرکت نکند. برای این منظور می‌توانیم در تعریف ماشین تورینگ گزینه‌ی ایست (Stay-option) را اضافه کنیم:

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$$

که در آن تعبیر S عدم حرکت هد خواندن/نوشتن است.
این گزینه قدرت ماشین تورینگ را افزایش نمی‌دهد.

قضیه کلاس ماشین‌های تورینگ با گزینه‌ی ایست و کلاس ماشین‌های تورینگ استاندارد، هم‌ارز هستند.

◄ **اثبات.**
از آنجا که ماشین تورینگ با گزینه‌ی ایست به طور واضح توسعه‌ای از مدل استاندارد است، بدیهی است که هر ماشین تورینگ استاندارد می‌تواند با یک ماشین با گزینه‌ی ایست شبیه‌سازی شود.
برای نشان دادن عکس این مطلب فرض می‌کنیم

$$M = (Q, \Sigma, \Gamma, \delta, q_\circ, \square, F)$$

یک ماشین تورینگ با گزینه‌ی ایست باشد که باید با ماشین تورینگ استاندارد $\hat{M} = (\hat{Q}, \Sigma, \Gamma, \hat{\delta}, \hat{q}_\circ, \square, \hat{F})$ شبیه‌سازی شود.
برای هر حرکت M، ماشین شبیه‌سازی‌کننده‌ی $\hat{M}$ عمل زیر را انجام می‌دهد:

- اگر حرکت M حاوی گزینه‌ی ایست نباشد، ماشین شبیه‌سازی‌کننده حرکتی را انجام می‌دهد که اساساً مشابه حرکتی است که باید شبیه‌سازی شود.
- اگر S در حرکت M وجود داشته باشد، آنگاه $\hat{M}$ دو حرکت انجام می‌دهد:

۱) حرکت اول، آن نماد را بازنویسی می‌کند و هد را به سمت راست حرکت می‌دهد.

۲) حرکت دوم، هد خواندن/نوشتن را به سمت چپ حرکت می‌دهد و محتوای نوار را بدون تغییر باقی می‌گذارد.

ماشین شبیه‌سازی‌کننده می‌تواند از روی M با تعریف $\hat{\delta}$ به صورت زیر ساخته شود:
برای هر گذر

$$\delta(q_i, a) = (q_j, b, L \text{ or } R)$$

در $\hat{\delta}$ قرار می‌دهیم:

$$\delta(\hat{q_i}, a) = (\hat{q}_j, b, L \text{ or } R)$$

و برای گذر S

$$\delta(q_i, a) = (q_j, b, S)$$

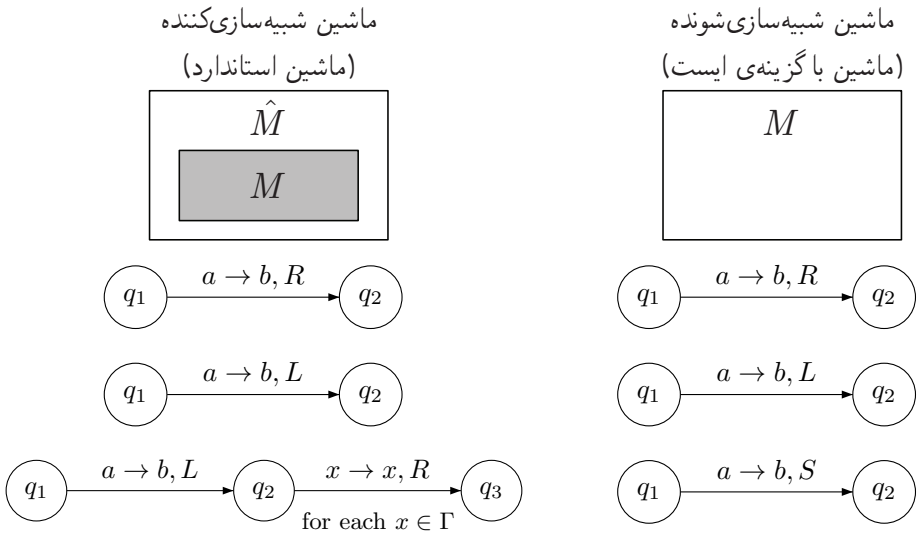
گذرهای متناظر زیر را در $\hat{\delta}$ قرار می‌دهیم:

$$\hat{\delta}(\hat{q}_i, a) = (\hat{q}_{js}, b, R)$$

و

$$\hat{\delta}(\hat{q}_{js}, c) = (\hat{q}_j, c, L) \qquad , \forall c \in \Gamma$$

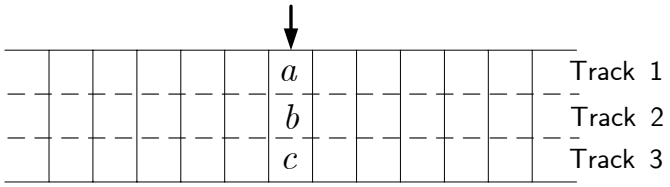
به طور منطقی واضح است که هر محاسبه‌ی M دارای محاسبه‌ی متناظری در $\hat{M}$ است، پس $\hat{M}$ می‌تواند M را شبیه‌سازی کند.



شبیه‌سازی، یک تکنیک استاندارد برای نشان دادن هم‌ارزی آتوماتاست.
برای صحبت در مورد این فرآیند به طور دقیق و اثبات قضایای هم‌ارزی می‌توان از روش رسمی توصیف شده استفاده کرد.
در ادامه‌ی این بحث از مفهوم شبیه‌سازی به دفعات استفاده خواهیم کرد، اما اغلب سعی نمی‌کنیم چیزی را به طور دقیق و جزیی توصیف کنیم. شبیه‌سازی کامل با ماشین‌های تورینگ اغلب خسته کننده است.
برای همین، بحث را به جای شکل اثبات قضیه به صورت توصیفی انجام می‌دهیم.
شبیه‌سازی‌ها تنها در یک طرح کلی ارائه می‌شوند، اما دیدن اینکه چگونه می‌توانند دقیق شوند، نباید چندان سخت باشد.
این توصیفات می‌توانند برای طراحی شبیه‌سازی در یک زبان سطح بالاتر یا شبه کد آموزنده باشند.

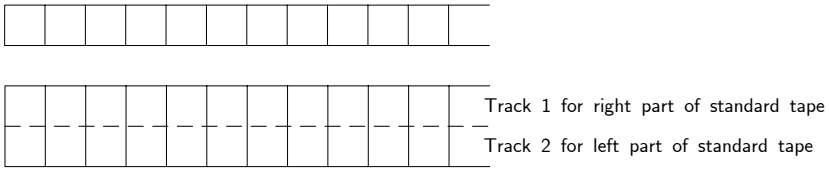
۱۰-۱-۳ ماشین تورینگ چندشیاره

ماشین تورینگ چندشیاره (multiple-track) همان ماشین تورینگ استاندارد است، با این تفاوت که الفبای مورد استفاده به صورت kتایی‌های مرتب از الفبای نوار ماشین استاندارد است (می‌توان تصور کرد که هر نماد از چند بخش تشکیل شده است).
به عبارت دیگر، هر سلول نوار به k بخش (شیار) تقسیم می‌شود.



۱۰-۱-۴ ماشین تورینگ با نوار نیمه نامتناهی

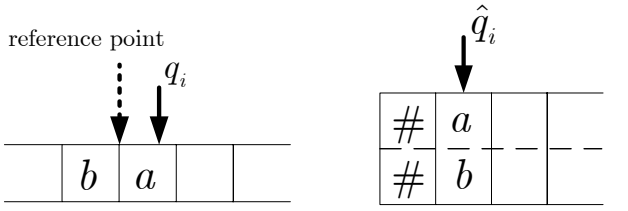
در ماشین تورینگ با نوار نیمه نامتناهی (semi-infinite tape)نوار از یک سر متناهی است، در واقع نوار را در این ماشین از وسط تا کرده‌ایم.
ماشین تورینگ با نوار نیمه نامتناهی مشابه ماشین استاندارد است، با این تفاوت که وقتی هد خواندن / نوشتن در مرز قرار دارد هیچ حرکتی به سمت چپ مجاز نیست.
چندان دشوار نیست که ببینیم این محدودیت تاثیری بر قدرت ماشین تورینگ نمی‌گذارد.
برای شبیه‌سازی یک ماشین تورینگ استاندارد M با ماشین $\hat{M}$ با نوار نیمه نامتناهی می‌توان مطابق شکل زیر عمل کرد:



ماشین شبیه‌سازی کننده‌ی $\hat{M}$ یک نوار با دو شیار دارد. در شیار بالایی اطلاعات مربوط به سمت راست یک نقطه‌ی مرجع بر روی نوار M قرار دارد.
نقطه‌ی مرجع مثلاً می‌تواند مکان هد خواندن / نوشتن در شروع محاسبه باشد.
شیار پایینی حاوی بخش سمت چپ نوار M به ترتیب معکوس است. $\hat{M}$ به گونه‌ای برنامه‌ریزی می‌شود که اطلاعات مربوط به شیار بالایی را تنها وقتی استفاده کند که هد خواندن / نوشتن M در سمت راست نقطه‌ی مرجع باشد و شیار پایینی را تنها هنگامی استفاده کند که M در سمت چپ نوار خود حرکت می‌کند.
برای ایجاد این تمایز می‌توان مجموعه‌ی حالات $\hat{M}$ را به دو بخش Q_U و Q_L افراز کرد:

- Q_U وقتی استفاده می‌شود که روی شیار بالایی کار می‌کنیم.
- Q_L وقتی استفاده می‌شود که روی شیار پایینی کار می‌کنیم.

در واقع در ماشین شبیه‌سازی‌کننده، از حالت‌های ماشین شبیه‌سازی‌شونده دو کپی تهیه می‌کنیم و یکی را با اندیس U و دیگری را با اندیس L نامگذاری می‌کنیم.
نماد پایان خاص # برر روی مرز چپ نوار قرار می‌گیرد تا سوییچ کردن از یک شیار به شیار دیگر را ساده نماید.
برای مثال فرض کنید ماشینی که باید شبیه‌سازی شود و ماشین شبیه‌سازی کننده در پیکربندی‌های متناظر نشان داده شده در شکل زیر قرار داشته باشند:



ماشینی که باید شبیه‌سازی شود
Machine to be simulated

ماشین شبیه‌سازی کننده
Simulating machine

و حرکتی که باید شبیه‌سازی شود با

$$\delta(q_i, a) = (q_j, c, L)$$

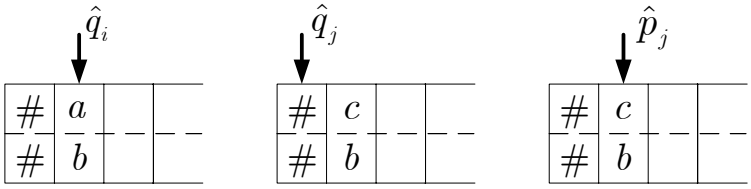
تولید شود. ماشین شبیه‌سازی کننده ابتدا با گذر

$$\hat{\delta}(\hat{q}_i, (a, b)) = (\hat{q}_j, (c, b), L)$$

حرکت می‌کند که در آن $\hat{q}_i \in Q_U$. از آنجا که $\hat{q}_i \in Q_U$ است، تنها اطلاعات در شیار بالایی این نقطه در نظر گرفته می‌شود. اکنون ماشین شبیه‌سازی کننده $(\#, \#)$ را در حالت $\hat{q}_j \in Q_U$ می‌بیند سپس از یک گذر به صورت

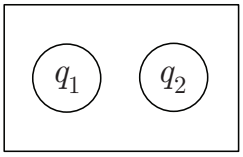
$$\hat{\delta}(\hat{q}_i, (\#, \#)) = (\hat{p}_j, (\#, \#), R)$$

استفاده می‌کند که در آن $\hat{p}_j \in Q_L$ است و آن را در پیکربندی متناظر با شکل زیر قرار می‌دهد:

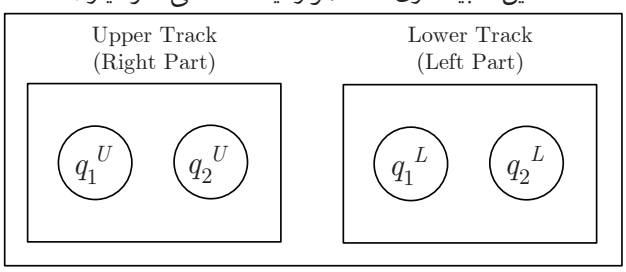


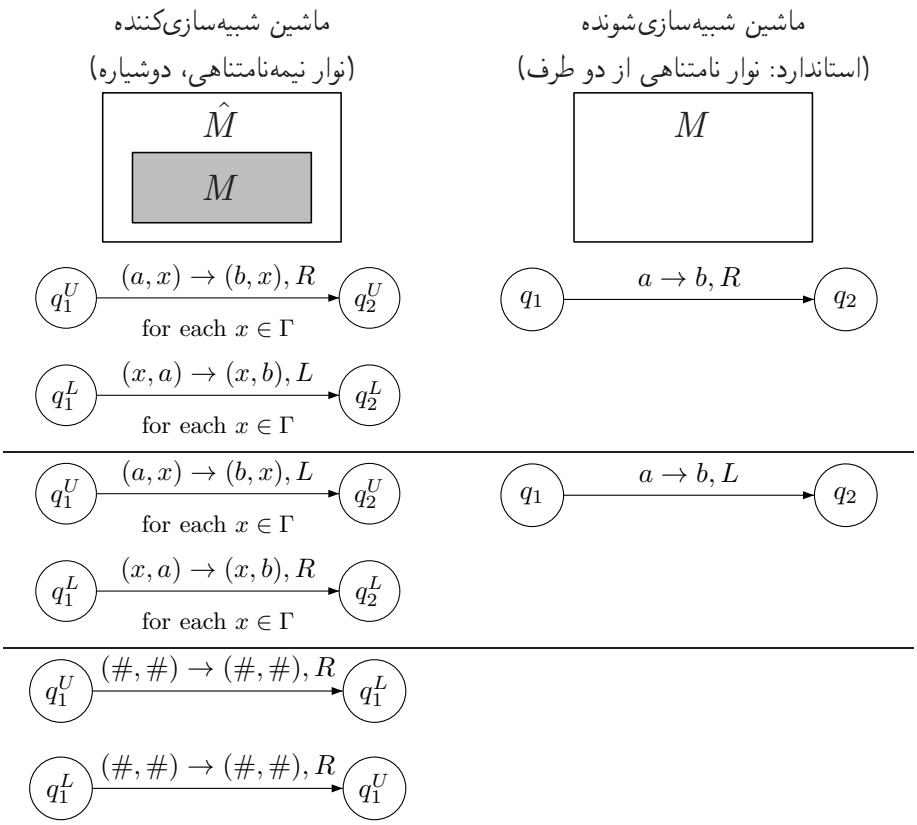
جزییات بیشتر این شبیه‌سازی سرراست است.

ماشین شبیه‌سازی‌شونده (استاندارد: نوار نامتناهی از دو طرف):



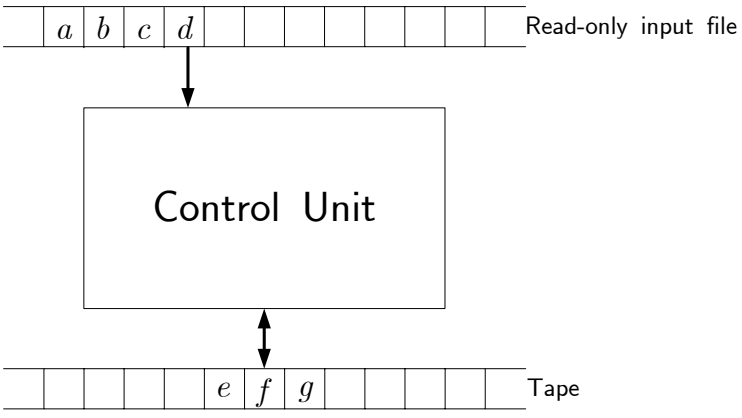
ماشین شبیه‌سازی‌کننده (نوار نیمه‌نامتناهی، دوشیاره):





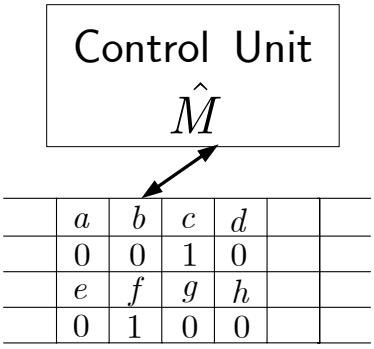
۵-۱-۱۰ ماشین تورینگ برون‌خطی

ماشین تورینگ برون‌خطی (offline) علاوه بر نوار، یک فایل ورودی هم دارد که مانند گذشته فقط قابل خواندن است.



رفتار هر ماشین تورینگ استاندارد می‌تواند با یک مدل برون‌خطی شبیه‌سازی شود.
چیزی که باید توسط ماشین شبیه‌سازی کننده انجام شود، کپی کردن ورودی از فایل ورودی به نوار است.
از این پس آن را به همان روش ماشین استاندارد پردازش می‌کنیم.
شبیه‌سازی ماشین برون‌خطی M با ماشین استاندارد $\hat{M}$ به توصیف طولانی‌تری نیاز دارد. یک ماشین استاندارد می‌تواند محاسبات یک ماشین برون‌خطی را با استفاده از چینش چهارشیاره مطابق شکل زیر

شبیه‌سازی کند. در این شکل محتوای نوار برای یک پیکربندی خاص از شکل بالا نشان داده شده است:



هر یک از چهار شیار $\hat{M}$ یک نقش خاص در شبیه‌سازی بازی می‌کند:

- شیار اول ورودی را در بر دارد.
- شیار دوم مکان ورودی خوانده شده را علامت‌گذاری می‌کند.
- شیار سوم نوار M را بازنمایی می‌کند.
- شیار چهارم مکان هد خواندن / نوشتن M را بازنمایی می‌کند.

شبیه‌سازی هر حرکت M نیاز به تعدادی حرکت $\hat{M}$ دارد:

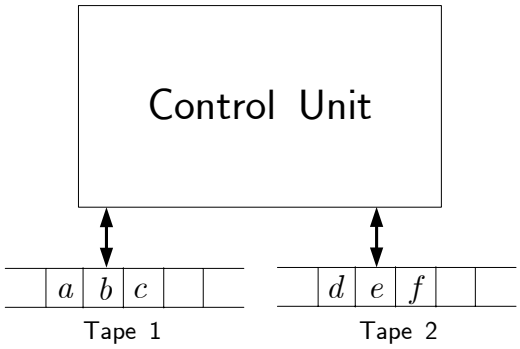
- با شروع از یک موقعیت استاندارد، مثلاً انتهای چپ، و با اطلاعات مربوط علامت‌گذاری شده با علامت‌گذارهای مخصوص انتها، $\hat{M}$ شیار دوم را برای پیدا کردن موقعیت خوانده شده از فایل ورودی M را جستجو می‌کند.
- نماد یافت شده در سلول متناظر بر روی شیار اول با قرار دادن واحد کنترل $\hat{M}$ در حالت انتخاب شده برای این منظور به خاطر آورده می‌شود.
- سپس شیار چهارم به دنبال موقعیت هد خواندن / نوشتن M جستجو می‌شود.
- با ورودی به خاطر آورده شده و نماد روی شیار سوم، ما اکنون می‌دانیم که M باید چه کاری انجام دهد.
- این اطلاعات از طریق یک حالت داخلی مناسب دوباره توسط $\hat{M}$ به خاطر آورده می‌شود.
- سپس هر چهار شیار نوار $\hat{M}$ تغییر داده می‌شود تا حرکت M را منعکس نماید. در نهایت، هد خواندن / نوشتن $\hat{M}$ برای شبیه‌سازی حرکت بعدی به موقعیت استاندارد برمی‌گردد.

۱۰-۲ ماشین تورینگ با حافظه‌ی پیچیده‌تر

ساختار حافظه‌ی ماشین تورینگ استاندارد آن قدر ساده است که ممکن است تصور شود بتوان با استفاده از ساختار حافظه‌ی پیچیده‌تر قدرت آن را افزایش داد، اما در ادامه مشاهده خواهیم کرد که این طور نیست.

۱۰-۲-۱ ماشین تورینگ چندنواره

در ماشین تورینگ چندنواره (multiple-tape) هر یک از نوارها دارای هد خواندن / نوشتن مختص خود هستند.



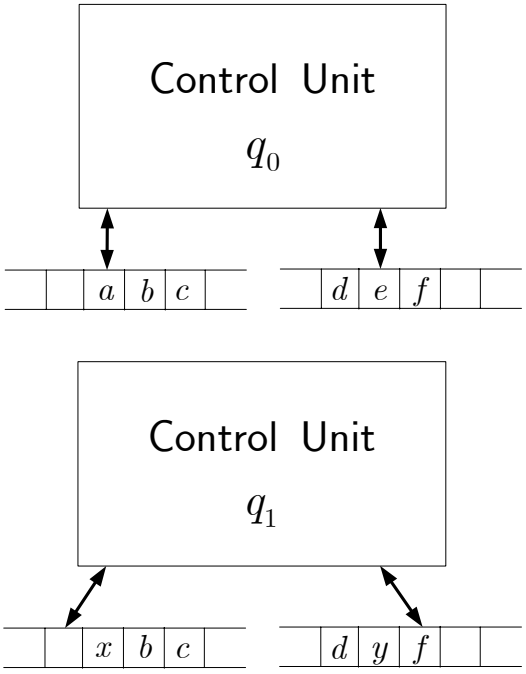
تعریف رسمی این ماشین مشابه ماشین تورینگ استاندارد است که در آن تابع گذر به صورت

$$\delta : Q \times \Gamma^n \to Q \times \Gamma^n \times \{L, R\}^n$$

تعریف می‌شود و هر آنچه برای هر n نوار رخ می‌دهد را نشان می‌دهد.
قدرت این ماشین از ماشین تورینگ معمولی بیشتر نیست.

مثال

برای $n = ۲$ پیکربندی‌های نشان داده شده در شکل زیر



و قاعده‌ی گذر

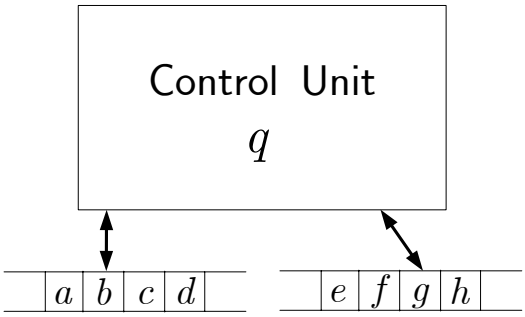
$$\delta(q_۰, a, e) = (q_۱, x, y, L, R)$$

را درنظر می‌گیریم. قاعده‌ی گذر در صورتی اعمال می‌شود که ماشین در حالت $q_۰$ باشد و هد خواندن/نوشتن اول، a و هد خواندن/ نوشتن دوم، e را ببیند. نماد روی نوار اول با x جایگزین می‌شود و هد خواندن/

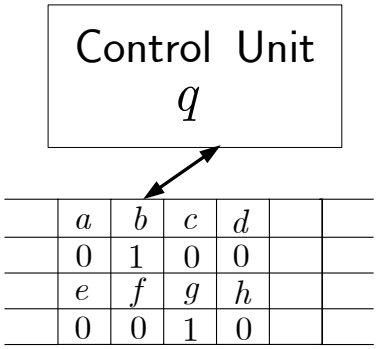
نوشتن آن به سمت چپ جابجا می‌شود. به طور همزمان، نماد روی نوار دوم با y بازنویسی می‌شود و هد خواندن/ نوشتن به سمت راست جابجا می‌شود. واحد کنترل حالت خود را به q_1 تبدیل می‌کند.

▲

برای نشان دادن هم‌ارزی بین ماشین تورینگ استاندارد و ماشین تورینگ چندنواره باید نشان دهیم که هر ماشین تورینگ چند نواره‌ی M می‌تواند با یک ماشین تورینگ استاندارد $\hat{M}$ شبیه‌سازی شود و برعکس هر ماشین تورینگ استاندارد می‌تواند با یک ماشین چندنواره شبیه‌سازی شود.
بخش دوم این ادعا نیاز به هیچ تلاشی ندارد، زیرا همیشه می‌توانیم در به کارگیری ماشین تورینگ چندنواره فقط از یک نوار آن برای انجام کار مفید استفاده کنیم.
شبیه‌سازی یک ماشین چندنواره با یک نوار اندکی پیچیده‌تر است، اما به لحاظ مفهومی سرراست است.
برای مثال یک ماشین دونواره را در نظر بگیرید که در پیکربندی مشخص شده در شکل زیر قرار دارد:



ماشین شبیه‌سازی کننده‌ی تک‌نواره‌ی $\hat{M}$ مطابق شکل زیر دارای ۴ شیار است:



شیار اول، محتوای نوار ۱ مربوط به M را نشان می‌دهد.
بخش غیرخالی شیار دوم تمام صفر است، بجز برای یک سلول که مقدارش ۱ است و موقعیت هد خواندن/ نوشتن M را علامت‌گذاری می‌کند.
شیار سوم و چهارم، نقش مشابهی را برای نوار ۲ مربوط به M بر عهده دارند.
بازنمایی یک ماشین چندنواره با یک ماشین تک‌نواره شبیه روش مورد استفاده در شبیه‌سازی یک ماشین برون‌خطی است. گام‌های واقعی در شبیه‌سازی نیز بسیار مشابه هستند، تنها تفاوت آنها این است که در اینجا تعداد نوار بیشتری باید در نظر گرفته شود.
طرح ارائه شده برای شبیه‌سازی ماشین‌های برون‌خطی در این مورد با تغییراتی جزیی اعمال می‌شود و روالی را پیشنهاد می‌دهد که با آن تابع گذر $\hat{\delta}$ مربوط به $\hat{M}$ بتواند از روی تابع گذر δ مربوط به M ساخته شود.

با وجود اینکه دقیق کردن این ساختمان دشوار نیست، اما نیاز به حجم زیادی نوشتن دارد. مطمئناً محاسبات روی $\hat{M}$ طولانی و خسته‌کننده است، اما این تاثیری بر بحث ما ندارد. آنچه می‌توانیم بر روی M انجام دهیم، بر روی $\hat{M}$ هم می‌توانیم انجام دهیم.

◄ **نکته** هنگامی که ادعا می‌کنیم یک ماشین تورینگ با نوارهای چندگانه قدرتمندتر از یک ماشین تورینگ استاندارد نیست، تنها در مورد آنچه می‌تواند توسط این ماشین‌ها انجام شود، یعنی زبانی که می‌پذیرند، حکم می‌کنیم و مسئله‌ی کارآمدی (زمان، فضا و . . .) مد نظر نیست.

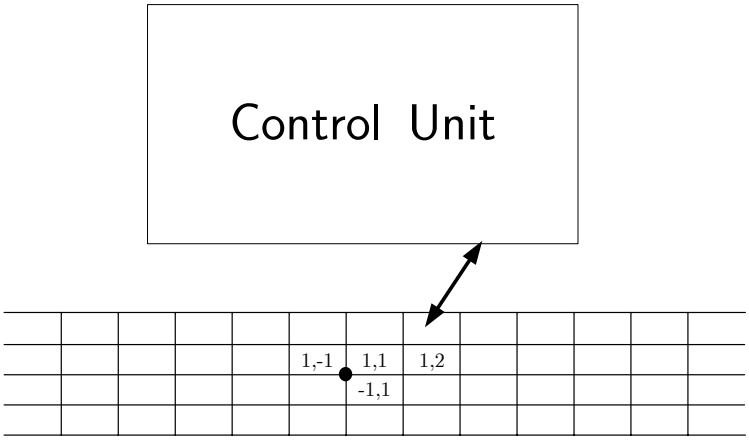
مثال

زبان $\{a^n b^n\}$ را در نظر بگیرید.
پذیرش این زبان با ماشین تورینگ تک نواره کار زیادی می‌طلبد.
اما با ماشین تورینگ دونواره این کار ساده‌تر و سریع‌تر می‌شود.
فرض کنید رشته‌ی اولیه‌ی $a^n b^n$ در ابتدای محاسبه روی نوار ۱ نوشته شده باشد؛
سپس همه‌ی aها را می‌خوانیم و آنها را در نوار ۲ کپی می‌کنیم.
وقتی به انتهای aها رسیدیم، bهای روی نوار ۱ را در مقابل aهای کپی شده روی نوار ۲ مطابقت می‌دهیم.
با روش می‌توان تعیین کرد که آیا تعداد مساوی a و b وجود دارد یا خیر، بدون اینکه لازم باشد هد خواندن/نوشتن به طور مکرر جلو عقب برده شود.

◄ **نکته** انواع مدل‌های ماشین تورینگ تنها نسبت به توانایی انجام کارها معادل در نظر گرفته می‌شوند، نه نسبت به سادگی برنامه‌ریزی یا هر معیار کارآمدی دیگر که ممکن است در نظر داشته باشیم.

۱۰-۲-۲ ماشین تورینگ چندبعدی

ماشین تورینگ چندبعدی (multi-dimensional) یک ماشین تورینگ است که در آن نوار در بیش از یک بعد به طور نامتناهی گسترش یافته است.



Two-dimensional address scheme

تعریف رسمی ماشین تورینگ دوبعدی شامل تابع گذر δ به صورت

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, U, D\}$$

است که در آن U و D جابجایی هد خواندن - نوشتن به بالا و پایین را مشخص می‌کند.
برای شبیه‌سازی این ماشین با یک ماشین تورینگ استاندارد، می‌توانیم از یک مدل دوشیاره مطابق شکل زیر استفاده کنیم:

	a				b								
	1	#	2	#	1	0	#	−	3	#			

ابتدا یک ترتیب یا آدرس به سلول‌های نوار دوبعدی نسبت می‌دهیم.
این کار می‌تواند به طرق متعددی انجام شود.
برای مثال، از دو شیار مربوط به نوار ماشین شبیه‌سازی کننده، یک شیار برای ذخیره‌ی محتوای سلول و دیگری برای نگهداری آدرس متناظر استفاده می‌شود.
مطابق شکل فوق، در این پیکربندی سلول $(۱, ۲)$ حاوی a و سلول $(۳-, ۱۰)$ حاوی b است.
در اینجا یک دشواری وجود دارد: آدرس سلول می‌تواند شامل اعداد صحیح به دلخواه بزرگ باشد، بنابراین شیار آدرس نمی‌تواند از یک فیلد با اندازه‌ی ثابت برای ذخیره کردن آدرس‌ها استفاده کند. به جای آن باید از آرایش فیلد با اندازه‌ی متغیر استفاده کنیم که در آن از نمادهای خاص برای جدا کردن فیلدها (مانند شکل فوق) استفاده می‌شود.
فرض می‌کنیم در ابتدای شبیه‌سازی هر حرکت، هد خواندن/ نوشتن ماشین دوبعدی M و هد خواندن/ نوشتن ماشین شبیه‌سازی کننده‌ی $\hat{M}$ همیشه بر روی سلول‌های متناظر باشد.
برای شبیه‌سازی یک حرکت، ابتدا ماشین شبیه‌سازی‌کننده‌ی $\hat{M}$ با استفاده از طرح آدرس‌دهی دوبعدی، آدرس سلولی که M باید بدان حرکت کند را با یک محاسبه‌ی ساده محاسبه می‌کند.
وقتی آدرس محاسبه شد، $\hat{M}$ سلول با این آدرس را روی شیار ۲ می‌یابد و سپس محتوای سلول را تغییر می‌دهد تا حرکت M شمرده شود.
مجدداً، با داشتن M ساختار سرراستی برای $\hat{M}$ وجود خواهد داشت.

۱۰-۳ ماشین تورینگ غیرقطعی

چون در تز تورینگ به نوع خاصی از نوار تاکید نشده است، پس تاثیری در قدرت ماشین ندارد، اما در مورد ماشین تورینگ غیرقطعی به راحتی نمی‌توان گفت که تفاوتی با نوع قطعی ندارد و باید بررسی بیشتری انجام شود.

تعریف یک ماشین تورینگ غیرقطعی، آتوماتونی است مطابق با تعریف ماشین تورینگ، با این تفاوت که تابع δبه صورت زیر تعریف می‌شود:

$$\delta : Q \times \Gamma \to ۲^{Q \times \Gamma \times \{L,R\}}$$

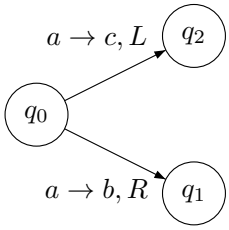
همانند همیشه وقتی عدم قطعیت مطرح است، برد δ مجموعه‌ای از تغییر وضعیت‌هاست که هر کدام از آنها می‌تواند توسط ماشین انتخاب شود.

مثال

اگر $\delta(q_0, a) = \{(q_1, b, R), (q_2, c, L)\}$ باشد، TM غیرقطعی است و حرکت‌های

$$q_0\ aaa \vdash b\ q_1\ aa, \quad q_0\ aaa \vdash q_2\ caa$$

هر دو ممکن است.



چون نقش عدم قطعیت در محاسبه‌ی توابع واضح نیست، آتوماتای غیرقطعی معمولاً در نقش پذیرنده است.

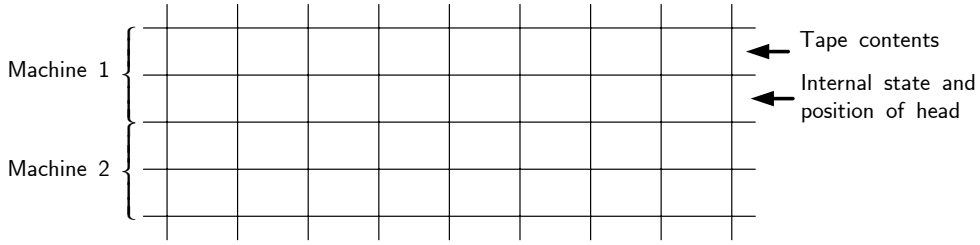
ماشین تورینگ غیرقطعی در صورتی رشته‌ی w را می‌پذیرد که دنباله‌ای از حرکات وجود داشته باشد که $q_0\ w \vdash^* x_1\ q_f\ x_2$ به طوری که $q_f \in F$.

یک ماشین غیرقطعی ممکن است دارای حرکاتی باشد که به یک حالت نهایی یا حلقه‌ی بی‌نهایت منجر شود، اما مانند گذشته این انتخاب‌ها بی‌اهمیت هستند. آنچه مهم است این است که دنباله‌ای از حرکات برای پذیرش وجود دارد.

برای این که نشان دهیم ماشین تورینگ غیرقطعی قدرتمندتر از ماشین تورینگ قطعی نیست، باید یک معادل قطعی برای آن ارائه دهیم.

عدم قطعیت می‌تواند به عنوان یک الگوریتم قطعی با عقب‌گرد (backtracking) در نظر گرفته شود $\Leftarrow$ یک ماشین قطعی می‌تواند یک ماشین غیرقطعی را شبیه‌سازی کند، اگر بتواند عمل عقب‌گرد را انجام دهد. برای اینکه ببینیم این کار چگونه قابل انجام است، باید به نحو دیگری به عدم قطعیت نگاه کنیم.

یک ماشین غیرقطعی ماشینی است که بتواند خود را تکرار کند. وقتی که بیش از یک حرکت امکان‌پذیر باشد، ماشین به تعداد لازم از خود کپی تهیه می‌کند و به هر کدام از کپی‌ها کاری را واگذار می‌کند که هر یک از آنها یکی از انتخاب‌های ماشین است. این نگرش به عدم قطعیت ممکن است غیر مکانیکی به نظر برسد، تکرار نامحدود قطعاً از قدرت کامپیوترهای امروزی خارج است، گرچه می‌تواند توسط یک ماشین تورینگ قطعی شبیه‌سازی شود:



یک ماشین تورینگ با نوار دوبعدی را در نظر می‌گیریم.
هر زوج شیار، یک ماشین را بازنمایی می‌کند.
شیار بالایی محتوای نوار و شیار پایینی محل هد را نشان می‌دهد.
وقتی که نیاز به یک ماشین جدید داریم، دو شیار جدید شروع می‌شود و اطلاعات مناسب در آن قرار داده می‌شود.

قضیه کلاس ماشین‌های تورینگ قطعی و کلاس ماشین‌های تورینگ غیرقطعی با هم برابر است.

۴-۱۰ ماشین تورینگ جامع

تعریفی که از ماشین تورینگ ارائه شد، یک کامپیوتر خاص منظوره است.
وقتی δ تعریف شود، ماشین محدود به انجام نوع خاصی از محاسبه می‌شود.
اما از طرف دیگر کامپیوترهای دیجیتال همه‌منظوره هستند و می‌توانند برای انجام کارهای مختلف برنامه‌ریزی شوند.
پس ماشین تورینگ نمی‌تواند معادل با کامپیوترهای همه‌منظوره باشد!
این انتقاد را می‌توان با طراحی یک ماشین تورینگ قابل برنامه‌ریزی مجدد با نام «ماشین تورینگ جامع» رفع نمود.

تعریف ماشین تورینگ جامع *(universal turing machine)*، M_u، آتوماتونی است که وقتی توصیف یک ماشین تورینگ و رشته‌ی w به عنوان ورودی به آن داده می‌شود، می‌تواند محاسبات M بر روی w را شبیه‌سازی کند.

برای ساخت M_u ابتدا روشی استاندارد برای توصیف ماشین تورینگ می‌یابیم. فرض می‌کنیم

$$Q = \{q_1, q_2, \ldots, q_n\}$$

با q_1 به عنوان حالت اولیه و q_2 به عنوان حالت نهایی و

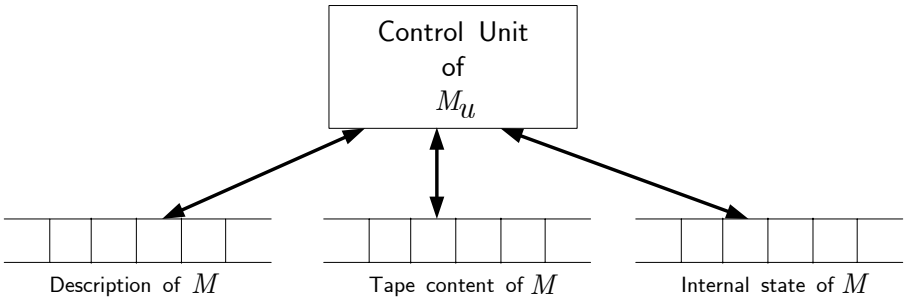
$$\Gamma = \{a_1, a_2, \ldots, a_m\}$$

که a_1 فاصله‌ی خالی است.
کدگذاری:
q_1 را با ۱، q_2 را با ۱۱، q_3 را با ۱۱۱، ... نمایش می‌دهیم.
a_1 را با ۱، a_2 را با ۱۱، a_3 را با ۱۱۱، ... نمایش می‌دهیم.
۰ برای جداسازی بین ۱ها استفاده می‌شود.
با حالت اولیه، نهایی و فاصله‌ی خالی که به صورت فوق تعریف شده‌اند، هر ماشین تورینگی به طور کامل توصیف می‌شود.

تابع تغییر حالت δ بر اساس همین روش و آرگومان‌ها و نتایج به یک ترتیب از پیش تعیین شده نوشته می‌شود، مثلاً:

$$\delta(q_۱, a_۲) = (q_۲, a_۳, L) \quad : \quad \ldots ۱۰۱۱۰۱۱۰۱۱۱۰۱۰ \ldots$$

در نتیجه هر ماشین تورینگ دارای یک کدگذاری متناهی به صورت رشته‌ای در $\{۰, ۱\}^+$ است.
هر کد از M را می‌توان به صورت منحصر به فرد کدگشایی کرد.
بعضی از رشته‌ها هیچ ماشین تورینگی را نمایش نمی‌دهند، اما به سادگی قابل تشخیص هستند، مانند ۰۰۰۱۱.
ماشین تورینگ جامع M_u دارای یک الفبای ورودی است که $\{۰, ۱\}$ را شامل می‌شود و دارای ساختار چندنواره است.



برای هر ورودی M و w یک تعریف کدگذاری شده از M را نگهداری می‌کنیم:
نوار ۲ حاوی محتویات نوار M و
نوار ۳ حاوی حالت‌های داخلی M است.
M_u ابتدا به محتوای نوارهای ۲ و ۳ نگاه می‌کند تا پیکربندی M را تعیین کند.
بعد به نوار ۱ مراجعه می‌کند تا ببیند که M در این پیکربندی چه کاری را انجام می‌دهد.
نهایتاً نوارهای ۲ و ۳ به گونه‌ای تغییر داده می‌شوند که نتیجه‌ی حرکت را منعکس نماید.

مجموعه‌های شمارا. برای مشاهده‌ی این که هر ماشین تورینگ را می‌توان با رشته‌ای از صفرها و یک‌ها نمایش دارد، باید چند نتیجه از نظریه‌ی مجموعه‌ها را مرور کنیم.
همه‌ی مجموعه‌های متناهی شمارا هستند.
در مجموعه‌های نامتناهی، برخی مجموعه‌ها **شمارا** هستند (مجموعه‌هایی که بتوان بین آنها و اعداد صحیح مثبت تناظر یک به یک برقرار کرد) و سایر مجموعه‌ها **ناشمارا** هستند.

مثال

مجموعه‌ی اعداد گویا شماراست.

اگر بتوانیم روشی برای مرتب کردن عناصر یک مجموعه داشته باشیم، می‌توان ثابت کرد که آن مجموعه شماراست. به این روش یک روال شمارش می‌گوییم.
چون شمارش یک فرآیند مکانیکی است، می‌توان از مدل ماشین تورینگ برای تعریف کلی آن استفاده کرد.

تعریف فرض کنید که S مجموعه‌ی رشته‌های الفبای Σ باشد. در این صورت یک روال شمارش برای S، ماشین تورینگی است که قدم‌های زیر را طی می‌کند:

$$q_\circ \ \square \vdash^* q_s \ x_1 \# s_1 \vdash^* q_s \ x_2 \# s_2 \dots$$

که $s_i \in S$ و $x_i \in \Gamma^* - \{\#\}$ به گونه‌ای است که هر $s \in S$ در طی قدم‌های متناهی تولید می‌شود. حالت q_s حالتی است که عضویت در S را نشان می‌دهد: یعنی هرگاه q_s رخ می‌دهد، رشته‌ی پس از $\#$ در S است.

هر مجموعه‌ای که برای آن یک روال شمارش موجود باشد، شماراست.
تکنیک روال شمارش را نمی‌توان «الگوریتم» نامید، زیرا اگر S نامتناهی باشد، متوقف نمی‌شود، اگرچه یک فرآیند معنی‌دار است چون نتایج خوش‌تعریف و قابل پیش‌بینی تولید می‌کند.

مثال

اگر $\Sigma = \{a, b, c\}$ باشد، Σ^+ شماراست (مرتب‌سازی با ترتیب سره proper order: مرتب‌سازی بر اساس طول رشته و سپس ترتیب حروف الفبا).

قضیه مجموعه‌ی تمامی ماشین‌های تورینگ، نامتناهی، اما شماراست.

◄ اثبات.
می‌توان هر ماشین تورینگ را با استفاده از ۰ و ۱ کدگذاری کرد. روال شمارش با این کدگذاری به صورت زیر است:

۱) رشته‌ی بعدی در $\{۰, ۱\}^+$ را به ترتیب سره تولید کنید.

۲) اگر این رشته یک ماشین تورینگ است، آن را بر روی نوار بنویسید و گرنه آن را نادیده بگیرید.

۳) به قدم (۱) برگردید.

چون هر ماشین تورینگ دارای توصیفی متناهی است، هر ماشینی در نهایت با این فرآیند متوقف می‌شود، پس یک ترتیب وجود دارد و این مجموعه شماراست.

۱۰-۵ آتوماتای کران‌دار خطی

ماشین تورینگ را به گونه‌ای تغییر می‌دهیم که فقط بتواند از قسمتی از نوار که ورودی در آن قرار دارد، استفاده کند. یک آتوماتون کران‌دار خطی (linear bounded automaton: LBA) مانند ماشین تورینگ استاندارد دارای نوار نامحدود است، اما مقداری از نوار که می‌تواند از آن استفاده کند، تابعی خطی از طول ورودی است، زیرا می‌توان برای نوار چند شیار در نظر گرفت:
قسمت قابل استفاده از نوار را به سلول‌هایی از نوار که حاوی ورودی است، محدود می‌کنیم.
برای حفاظت از این محدوده، از دو علامت مخصوص به نام نماد **انتهای چپ**] و **انتهای راست** [استفاده می‌کنیم.
برای یک ورودی مانند w، پیکربندی اولیه‌ی ماشین تورینگ با توصیف بلافصل $q_0\,[w]$ نشان داده می‌شود.
علامت‌های انتهای چپ و انتهای راست را نمی‌توان بازنویسی کرد و هد خواندن / نوشتن نمی‌تواند به سمت چپ] یا سمت راست [برود.
گاهی گفته می‌شود که هد خواندن / نوشتن به علامت انتها برخورد می‌کند.

تعریف آتوماتون کران‌دار خطی، یک ماشین تورینگ $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ مطابق تعریف است که الفبای آن باید دارای نمادهای [و] باشد، به طوری که:
$\delta(q_i, [)$ فقط می‌تواند حاوی عناصری به شکل $(q_j, [, R)$ و
$\delta(q_i,])$ فقط می‌تواند حاوی عناصری به شکل $(q_j,], L)$ باشد.

تعریف رشته‌ای می‌تواند توسط یک آتوماتون کران‌دار خطی پذیرفته شود که به ازای $q_f \in F$ و $x_1, x_2 \in \Gamma^*$، حرکت‌هایی به صورت

$$q_0\ [w] \vdash^* [x_1\ q_f\ x_2]$$

ممکن باشد.
زبان پذیرفته شده توسط این LBA، مجموعه‌ی همه‌ی چنین رشته‌های پذیرفته‌ای است.

در این تعریف LBA غیرقطعی فرض شده است. این کار برای راحتی در بحث‌ها نیست، بلکه در تعریف LBA ضروری است. در عین حال که می‌توان LBA قطعی را تعریف کرد، اما هنوز نمی‌دانیم که آیا با نسخه‌ی غیرقطعی هم‌ارز است یا خیر (مسئله‌ی باز).

مثال

زبان $L = \{a^n b^n c^n : n \geq 1\}$ توسط یک آتوماتون کران‌دار خطی پذیرفته می‌شود.
با توجه به ماشین تورینگ این زبان، محاسبات به فضایی خارج از ورودی نیاز ندارد، پس این کار می‌تواند توسط یک آتوماتون کران‌دار خطی انجام شود.

مثال

یافتن یک LBA برای $L = \{a^{n!} : n \geq 0\}$.

یک راه‌حل برای حل این مساله این است که aها را به طور پیوسته به ۲، ۳، ۴، ...تقسیم کنیم تا این که رشته قبول یا رد شود.

اگر ورودی در L باشد، نهایتاً فقط یک a باقی می‌ماند،

وگرنه، باقیمانده‌ی غیرصفری خواهیم داشت.

نوار یک LBA می‌تواند چند شیار داشته باشد و شیارهای اضافی را می‌توان به عنوان پیش‌نویس استفاده کرد.

برای حل این مساله می‌توان از یک نوار دوشیاره استفاده کرد:

شیار اول حاوی تعداد aهایی است که پس از تقسیم باقی می‌ماند و

شیار دوم حاوی مقسوم علیه فعلی است.

	[	a	a	a	a	a	a	]	a's to be examined
	[	a	a	a				]	current divisor

با استفاده از مقسوم علیه روی نوار دوم، aهای نوار اول را با حذف همه‌ی نمادها بجز آنهایی که در ضریبی از مقسوم‌علیه اشتراک دارند، حذف می‌کنیم.

پس از این یکی به مقسوم علیه اضافه می‌کنیم تا یک باقیمانده‌ی غیر صفر داشته باشیم یا فقط یک a باقی بماند.

دو مثال اخیر نشان می‌دهد که LBA باید از PDA قوی‌تر باشد؛ زیرا دو زبان فوق مستقل از متن نیستند.

مراجع

[1] P. Linz, **An Introduction to Formal Languages and Automata**, 5th Ed., Jones and Bartletts, 2012.

[2] M. Sipser, **Introduction to the Theory of Computation**, 3rd Ed., Cengage Learning, 2013.