# فصل هشتم

# ماشین‌های تورینگ و نظریه چرچ– تورینگ

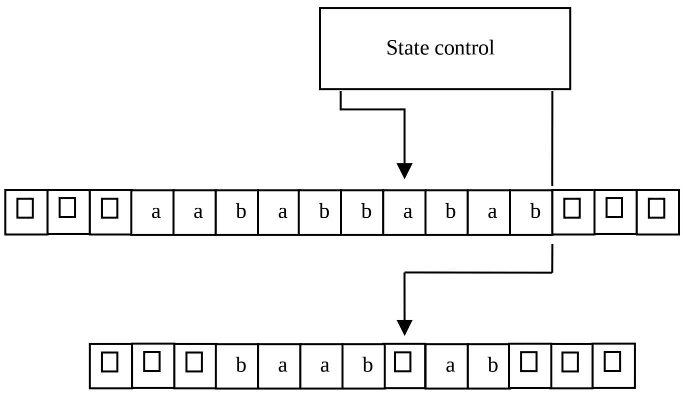
# (Turing Machines and the Church- Turing Thesis)

## 8-1: مقدمه

در فصل‌های پیش چند ابزار محاسباتی دیدیم که برای پذیرش یا تولید زبان‌های منظم و مستقل از متن می‌توانند بکار روند. با وجودی که این رده از زبان‌ها نسبتاً بزرگ هستند، اما دیدیم که این ابزارها به اندازه کافی قوی نیستند تا زبان‌های ساده‌ای نظیر $A = \{a^m b^n c^{mn} \mid m \geq ۰, n \geq ۰\}$ را بپذیرند. در این فصل ماشین تورینگ را معرفی می‌کنیم، این ماشین یک مدل ساده از کامپیوتر واقعی است. ماشین‌های تورینگ می‌توانند برای پذیرش همه زبان‌های مستقل از متن بکار روند، حتی زبان‌هایی نظیر A. این بحث را به میان می‌آوریم که هر مسأله‌ای که بتواند با یک کامپیوتر واقعی حل شود، همچنین با یک ماشین تورینگ نیز قابل حل است (این گزاره معروف است به نظریه چرچ- تورینگ). ماشین‌های تورینگ و از این روی، کامپیوترهای واقعی نیز دارای محدودیت‌هایی هستند که می‌توان درباره آن صحبت کرد.

## 8-2: تعریف یک ماشین تورینگ

موضوع را با توصیف غیر رسمی یک ماشین تورینگ آغاز می‌کنیم. چنین ماشینی شامل موارد زیر است (شکل زیر را هم ببینید).




**شکل 8-1:** ماشین تورینگ با نوارهای $k = ۲$ .

1. تعداد k نوار وجود دارد که k عددی ثابت و $k \geq 1$. هر نوار به سلول‌ها تقسیم شده است، و از هر دو طرف چپ و راست نامتناهی است. هر سلول یک نماد متعلق به یک مجموعه‌ی متناهی $\Gamma$ را در خود ذخیره می‌کند، این نماد را الفبای نوار گویند. الفبای نوار شامل نماد خالی یا □ است. اگر یک سلول شامل □ باشد، آنگاه این یعنی سلول عملاً خالی است.
2. هر نوار یک هد نوار دارد که می‌تواند در طول نوار حرکت نماید، در هر حرکت یک سلول را طی می‌کند. این هد می‌تواند سلولی را بخواند که به تازگی اسکن شده و نماد این سلول با نماد دیگر تعویض گردیده است.
3. یک واحد کنترل (حالت کنترل) وجود دارد که می‌تواند در هر یک از تعداد متناهی حالت‌ها باشد. مجموعه‌ی حالت‌ها را با Q نشان می‌دهیم. مجموعه‌ی Q شامل سه حالت خاص است، یک حالت شروع، یک حالت پذیرش و یک حالت رد کردن.

ماشین تورینگ دنباله‌ای از گام‌های محاسباتی را اجرا می‌کند. در یکی از چنین گام‌ها، موارد زیر را انجام می‌دهد:

1. بلافاصله پیش از گام محاسباتی، ماشین تورینگ در حالت r از Q قرار دارد، و هر یک از k هدهای نوار در سلولی قرار می‌گیرند.
2. بسته به حالت جاری r و k نمادی که توسط هد نوار خوانده شده‌اند،

(a) ماشین تورینگ به حالت r از Q می‌رود (که ممکن است مساوی r باشد).

(b) هر هد نوار یک نماد از $\Gamma$ را در سلولی که در حال اسکن شدن است، می‌نویسد (این نماد ممکن است با نمادی که اخیراً در سلول ذخیره شده است برابر باشد)، و

(c) هر هد نوار یا یک سلول به چپ حرکت می‌کند و یا یک سلول به راست می‌رود، و یا این که در سلول جاری می‌ایستد.

حال یک تعریف رسمی از ماشین تورینگ معین ارایه می‌کنیم.

**تعریف 8-2-1:** یک ماشین تورینگ معین عبارتست از یک هفت تایی

$$M = (\Sigma, \Gamma, Q, \delta, q, q_{accept}, q_{reject})$$

که در آن

1. $\Sigma$ یک مجموعه‌ی متناهی است که الفبای ورودی نامیده می‌شود، نماد خالی □ در $\Sigma$ قرار ندارد.
2. $\Gamma$ یک مجموعه‌ی متناهی است که الفبا نوار نامیده می‌شود. این الفبا نماد خالی □ را در بردارد و $\Sigma \subseteq T$،
3. $Q$ یک مجموعه‌ی متناهی است که عناصرش حالت‌ها نام دارند،
4. $q$ عضوی از $Q$ است و حالت شروع نام دارد.
5. $q_{accept}$ عضوی از $Q$ است، آن را حالت پذیرش گوییم.
6. $q_{reject}$ عضوی از $Q$ است، آن را حالت رد نامیم.
7. $\delta$ تابع انتقال است که در زیر مشخص شده است

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

تابع انتقال $\delta$ اساساً برنامه ماشین تورینگ است.

این تابع می‌گوید که ماشین در یک گام محاسباتی چه می‌کند؟

فرض کنیم $r \in Q$، و $a_۱, a_۲, ..., a_k \in \Gamma$. به علاوه گیریم $r' \in Q$ و $a'_۱, a'_۲, ..., a'_k \in \Gamma$ و $\sigma_۱, \sigma_۲, ..., \sigma_k \in \{L, R, N\}$ به طوری که $\delta(r, a_۱, a_۲, ..., a_k) = (r', a'_۱, a'_۲, ..., a'_k, \sigma_۱, \sigma_۲, ..., \sigma_۴)$

(8-1)

این انتقال بدین معنی است که اگر

- ماشین تورینگ در حالت r است و
- هد i امین نوار، نماد $a_i$ را بخواند که $۱ \leq i \leq k$

آنگاه

- ماشین تورینگ به حالت $r'$ منتقل می‌شود،
- هد مربوط به i امین نوار نماد اسکن شده $a_i$ را با نماد $a'_i$، $۱ \leq i \leq k$ جابه‌جا می‌کند و

- هد مربوط به $i$ امین نوار بر طبق $\delta_i$ که $1 \le i \le k$ حرکت می‌کند. اگر $\delta_1 = L$ آنگاه هد نوار یک سلول به چپ حرکت می‌کند، اگر $\delta_i = R$ ، آنگاه یک سلول به راست حرکت می‌نماید، اگر $\delta_i = N$ ، آنگاه هد نوار حرکت نمی‌نماید.

گام محاسباتی (1-8) را به صورت دستور زیر می‌نویسیم:

$$ra_1a_2...a_k \rightarrow r'a'_1a'_2...a'_k\delta_1\delta_2...\delta_4$$

حال محاسبات ماشین تورینگ را مشخص می‌کنیم.

$$M = (\Sigma, \Gamma, Q, \delta, q, q_{accept}, q_{reject})$$

## شروع شکل‌گیری

ورودی رشته‌ای است روی الفبای ورودی $\Sigma$ . در آغاز این رشته‌ی ورودی در اولین نوار ذخیره می‌شود و هد این نوار بر روی سمت چپ‌ترین نماد رشته‌ی ورودی قرار می‌گیرد. در شروع همه $k-1$ نوار دیگری تهی هستند، یعنی تنها نمادهای خالی را دارا هستند و ماشین تورینگ در حالت شروع $q$ می‌باشد.

## محاسبات و توقف

با آغاز شروع شکل‌گیری، ماشین تورینگ دنباله‌ای از گام‌های محاسباتی را، برابر آنچه در بالا توصیف شد، اجرا می‌کند. محاسبات در لحظه‌ای متوقف می‌شود که ماشین تورینگ حالت پذیرش یا $q_{accept}$ را وارد کند و یا این که حالت رد یا $q_{reject}$ را ارایه نماید (از این روی، اگر ماشین تورینگ هرگز حالت‌های $q_{accept}$ و $q_{reject}$ را وارد نکند، محاسبات متوقف نمی‌شود).

### پذیرش

ماشین تورینگ M رشته‌ی ورودی $w \in \Sigma^*$ را می‌پذیرد، اگر محاسبات بر روی این رشته در حالت $q_{accept}$ متوقف شود. اگر محاسبات بر روی این رشته در حالت $q_{reject}$ متوقف شود، آنگاه M رشته‌ی ورودی w را رد می‌کند.

زبان پذیرفته شده توسط ماشین تورینگ M را با L(M) نشان می‌دهیم. پس L(M) عبارتست از مجموعه‌ی همه‌ی رشته‌های داخل $\Sigma^*$ که توسط M پذیرفته شده‌اند. مشاهده شود که یک رشته‌ی $w \in \Sigma^*$ به L(M) تعلق ندارد، اگر و تنها اگر بر روی ورودی w:

- محاسبات M در حالت $q_{reject}$ متوقف شود یا
- محاسبات M متوقف نشود.

## 8-2-2: مثال‌هایی از ماشین‌های تورینگ

### مثال اول: پذیرش متقارن‌ها (Palindromes) با استفاده از یک نوار

نشان می‌دهیم چگونه یک ماشین تورینگ با یک نوار بسازیم که بتواند توصیف نماید آیا هر رشته‌ی ورودی $w \in \{a, b\}^*$ یک رشته‌ی متقارن است یا نه. یادآوری می‌کنیم که رشته‌ی w را یک رشته‌ی متقارن گویند، اگر خواندن w از چپ به راست با زمانی که از راست به چپ خوانده می‌شود، نتیجه یکسانی ارایه نماید. مثال‌هایی از متقارن‌ها عبارتند از: abba، baabbbbaab، و رشته تهی یا $\varepsilon$.

### شروع محاسبات

نوار شامل رشته‌ی ورودی w است، هر نوار در سمت چپ‌ترین نماد w قرار دارد و ماشین تورینگ در حالت شروع $q_0$ است.

**ایده:** هد نوار سمت چپ‌ترین نماد $w$ را می‌خواند، این نماد را حذف می‌کند و آن را به عنوان یک حالت بخاطر می‌سپارد. سپس هد نوار به سمت راست‌ترین نماد حرکت می‌کند و بررسی می‌کند که آیا آن با سمت چپ‌ترین نماد مساوی است یا خیر.

- اگر مساوی باشند، آنگاه سمت راست‌ترین نماد حذف می‌شود، هد نوار به سمت چپ‌ترین نماد حرکت می‌کند، و همه روند تکرار می‌گردد.
- اگر مساوی نباشند، ماشین تورینگ به حالت رد وارد می‌شود، و محاسبات متوقف می‌گردد.

ماشین تورینگ هر لحظه‌ای که رشته‌ی جاری و خیره شده بر روی نوار خالی باشد، به حالت پذیرش وارد می‌شود:

برای بررسی حالت‌های زیر، $\Sigma = \{a, b\}$ را الفبای ورودی و $\Gamma = \{a, b, \square\}$ را الفبای نوار در نظر می‌گیریم. مجموعه $Q$ از حالت‌ها، شامل هشت حالت زیر است:

$q_\circ$ : حالت شروع؛ هد نوار بر روی سمت چپ‌ترین نماد قرار دارد.

$q_a$ : سمت چپ‌ترین نماد، $a$ بوده است؛ هد نوار به راست حرکت می‌کند.

$q_b$ : سمت چپ‌ترین نماد، $b$ بوده است؛ هد نوار به راست حرکت می‌کند.

$q_a^{'}$ : سمت راست‌ترین نماد رسیده است؛ بررسی شود که اگر مساوی با $a$ است، آن را حذف کند.

$q_b^{'}$ : سمت راست‌ترین نماد رسیده است؛ بررسی شود که اگر مساوی با $b$ است، آن را حذف کند.

$q_L$ : بررسی مثبت بوده است؛ هد نوار به چپ حرکت می‌کند.

$q_{accept}$ : حالت پذیرش.

$q_{reject}$ : حالت رد (عدم پذیرش).

توابع انتقال $\delta$ با دستورات زیر مشخص می‌شود.

$q_0 a\square \to q_0 aaRR$

$q_0 b\square \to q_0 bbRR$

$q_0 \square\square \to q_1 \square\square LL$

$q_1 aa \to q_1 aaLN$

$q_1 ab \to q_1 abLN$

$q_1 ba \to q_1 baLN$

$q_1 bb \to q_1 bbLN$

$q_1 \square a \to q_2 \square aRN$

$q_1 \square b \to q_2 \square bRN$

$q_1 \square\square \to q_{accept}$

$q_2 aa \to q_2 aaRL$

$q_2 ab \to q_{reject}$

$q_2 ba \to q_{reject}$

$q_2 bb \to q_2 bbRL$

$q_2 \square\square \to q_{accept}$

**مثال دوم: پذیرش $a^n b^n c^n$ با استفاده از یک نوار**

یک ماشین تورینگ با یک نوار می‌سازیم که زبان زیر را پذیرش کند:

$\{a^n b^n c^n \mid n \geq 0\}$.

یادآوری می‌کنیم که این زبان مستقل از متن نیست.

**شروع محاسبات:** نوار شامل رشته‌ی ورودی $w$ است و هد نوار در سمت چپ‌ترین نماد $w$ قرار دارد. ماشین تورینگ در حالت شروع است.

**ایده:** در مثال‌های قبل، نوار الفبای $\Gamma$ با اجتماع الفبای ورودی $\Sigma$ و $\{\square\}$ برابر بود. در این مثال، یک نماد دیگر $d$ را به الفبای نوار می‌افزاییم. خواهیم دید که این امر ساختن ماشین تورینگ را ساده‌تر می‌سازد.

روال کلی عبارتست از تکرار موارد زیر:

در طول رشته از چپ به راست حرکت می‌کنیم و سمت چپ‌ترین $a$ را با $d$، سمت چپ‌ترین $b$ را با $d$، سمت چپ‌ترین $c$ را با $d$ جایگزین می‌نماییم و به سمت چپ‌ترین نماد برمی‌گردیم.

الفبای ورودی برابر $\Sigma = \{a, b, c\}$ و الفبای نوار $\Gamma = \{a, b, c, d, \square\}$ می‌باشند. حالت‌های زیر را بکار می‌بریم:

$q_a$ : حالت شروع، جست‌جو برای سمت چپ‌ترین $a$،

$q_b$ : سمت چپ‌ترین $a$ با $d$ جایگزین شده است، جست‌جو برای سمت چپ‌ترین $b$،

$q_c$ : سمت چپ‌ترین $a$ با $d$ و سمت چپ‌ترین $b$ با $d$ جایگزین شده‌اند، جست‌جو برای سمت چپ‌ترین $c$،

$q_L$ : سمت چپ‌ترین $a$ با $d$، سمت چپ‌ترین $b$ با $d$، و سمت چپ‌ترین $c$ با $d$ جایگزین شده‌اند، حرکت به سوی سمت چپ‌ترین نماد.

$q_{accept}$ : حالت پذیرش،

$q_{reject}$ : حالت رد (عدم پذیرش).

توابع انتقال $\delta$ با دستورات زیر مشخص می‌شود:

$$q_a a \rightarrow q_b dR$$
$$q_a b \rightarrow q_{reject}$$
$$q_a c \rightarrow q_{reject}$$
$$q_a d \rightarrow q_a dR$$
$$q_a \square \rightarrow q_{accept}$$

$$q_b a \rightarrow q_b aR$$
$$q_b b \rightarrow q_c dR$$
$$q_b c \rightarrow q_{reject}$$
$$q_b d \rightarrow q_b dR$$
$$q_b \square \rightarrow q_{reject}$$

$$q_c a \rightarrow q_{reject}$$
$$q_c b \rightarrow q_c bR$$
$$q_c c \rightarrow q_L dL$$
$$q_c d \rightarrow q_c dR$$
$$q_c \square \rightarrow q_{reject}$$

$$q_L a \rightarrow q_L aL$$
$$q_L b \rightarrow q_L bL$$
$$q_L c \rightarrow q_L cL$$
$$q_L d \rightarrow q_L dL$$
$$q_L \square \rightarrow q_a \square R$$

**مثال سوم: پذیرش $a^n b^n c^n$ با استفاده از الفبای نواری $\{a, b, c, \square\}$**

بار دیگر زبان $\{a^n b^n c^n \mid n \geq 0\}$ را در نظر می‌گیریم. در بخش پیش یک ماشین تورینگ ارایه کردیم که نماد اضافی $d$ را بکار برد. حال ممکن است خواننده فکر کند که آیا می‌توان یک ماشین تورینگ برای این زبان ساخت که هیچ نماد اضافی را بکار نبرد. نشان می‌دهیم که چنین امری واقعاً امکان پذیر است، به ترتیب زیر:

**شروع محاسبات:** نوار شامل رشته‌ی ورودی $w$ است و هد نوار بر روی سمت چپ‌ترین نماد $w$ قرار دارد. ماشین تورینگ در حالت شروع $q_0$ است.

**ایده:** مراحل 1 و 2 را تکرار کند تا زمانی که رشته تهی شود.

**مرحله 1.** در طول رشته از چپ به راست حرکت کند، سمت چپ‌ترین $a$، سمت چپ‌ترین $b$ و سمت راست‌ترین $c$ را حذف کند.

**مرحله 2.** زیر رشته‌های bها و cها را یک مکان به راست ببرد، آنگاه به سمت چپ‌ترین نماد برگردد.

الفبای ورودی و الفبای نوار به ترتیب عبارتند از $\Sigma=\{a,b,c\}$ و $\Gamma=\{a,b,c,d,\square\}$ . برای مرحله 1، حالت‌های زیر را بکار می‌بریم:

$q_0$ : حالت شروع؛ هد نوار در سمت چپ‌ترین نماد قرار دارد.

$q_a$ : سمت چپ‌ترین a حذف شده و b خوانده نشده است.

$q_b$ : سمت چپ‌ترین b حذف شده و c خوانده نشده است.

$q_c$ : سمت چپ‌ترین c خوانده شده است، و هد نوار به سمت راست حرکت می‌کند.

$q_c'$ : هد نوار به سمت راست‌ترین c قرار دارد.

$q_1$ : سمت راست‌ترین c حذف شده است، و هد نوار بر روی سمت راست‌ترین نماد و یا □ قرار دارد.

$q_{accept}$ : حالت پذیرش،

$q_{reject}$ : حالت رد (عدم پذیرش).

توابع انتقال مرحله‌ی 1 با دستورات زیر مشخص می‌شود:

| | |
|---|---|
| $q_0a \rightarrow q_0aR$ | $q_0a \rightarrow q_a\square R$ |
| $q_0b \rightarrow q_b\square R$ | $q_0b \rightarrow q_{reject}$ |
| $q_0c \rightarrow q_{reject}$ | $q_0c \rightarrow q_{reject}$ |
| $q_a\square \rightarrow q_{reject}$ | $q_0\square \rightarrow q_{accept}$ |

$q_b a \to q_{reject}$

$q_b b \to q_b bR$

$q_b c \to q_c cR$

$q_b \square \to q_{reject}$

$q_c a \to q_{reject}$

$q_c b \to q_{reject}$

$q_c c \to q_c cR$

$q_c \square \to q_c' \square L$

$q_c' c \to q_1 \square L$

برای مرحله‌ی 2، حالت‌های زیر را بکار می‌بریم:

$q_1$ : مانند بالا؛ هد نوار بر روی سمت راست‌ترین نماد یا بر روی □ قرار می‌گیرد.

$q^c$: c را در یک سلول به چپ کپی می‌کند.

$q^b$: b را در یک سلول به چپ کپی می‌کند.

$q_2$ : تغییر مکان‌ها انجام گیرد، هد به چپ حرکت می‌کند.

علاوه بر این، یک حالت $q_1'$ را که دارای معنی زیر است، بکار می‌بریم:

اگر رشته‌ی ورودی به فرم $a^i bc$ ، برای i ای که $i \ge 1$ ، باشد، آنگاه بعد از مرحله‌ی 1، نوار شامل رشته‌ی $a^{i-1}\square\square$ است، هد نوار بر روی □ ای که بلافاصله در راست a ها واقع است، قرار می‌گیرد، و ماشین تورینگ در حالت $q_1$ است. در این حال، یک سلول به چپ حرکت می‌کنیم؛ اگر □ را بخوانیم، آنگاه $i = 1$ ، و پذیرش انجام می‌شود، در غیر این صورت، a را می‌خوانیم و عدم پذیرش انجام می‌گیرد. توابع انتقال برای مرحله‌ی 2 با دستورات زیر مشخص می‌شود:

$q_1 a \to$ نمی‌تواند اتفاق بیفتد

$q_1 b \to q_{reject}$

$q_1 c \to q^c \square L$

$q_1 \square \to q_1' \square L$

$q_1' a \to q_{reject}$

$q_1' b \to$ نمی‌تواند اتفاق بیفتد

$q_1' c \to$ نمی‌تواند اتفاق بیفتد

$q_1' \square \to q_{accept}$

$q^b a \rightarrow$ نمی‌تواند اتفاق بیفتد

$q^b b \rightarrow q^b bL$

$q^b c \rightarrow$ نمی‌تواند اتفاق بیفتد

$q^b \square \rightarrow q_2 bL$

$q^c a \rightarrow$ نمی‌تواند اتفاق بیفتد

$q^c b \rightarrow q^b cL$

$q^c c \rightarrow q^c cL$

$q^c \square \rightarrow q_{reject}$

$q_2 a \rightarrow q_2 aL$

$q_2 b \rightarrow$ نمی‌تواند اتفاق بیفتد

$q_2 c \rightarrow$ نمی‌تواند اتفاق بیفتد

$q_2 \square \rightarrow q_0 \square R$

**مثال چهارم: پذیرش $a^m b^n c^{mn}$ با استفاده از یک نوار**

نشان می‌دهیم که چگونه یک ماشین تورینگ بسازیم که با یک نوار زبان زیر را پذیرش کند:

$$\{a^m b^n c^{mn} \mid m \geq 0, n \geq 0\}$$

یادآوری می‌کنیم که این زبان مستقل از متن نیست.

الفبای ورودی و الفبای نوار به ترتیب عبارتند از: $\Sigma = \{a, b, c\}$ و $\Gamma = \{a, b, c, \$, \square\}$. استفاده از نماد $\$$ را در زیر توضیح می‌دهیم.

**شروع محاسبات:** نوار شامل رشته‌ی ورودی w است و هد نوار در سمت چپ‌ترین نماد رشته‌ی w قرار دارد. ماشین تورینگ در حالت شروع می‌باشد.

**ایده:** مشاهده کنید که یک رشته‌ی $a^m b^n c^k$ در زبان قرار دارد، اگر و تنها اگر برای هر a، رشته شامل n تاc باشد، بر پایه این موارد، محاسبات شامل مراحل زیر خواهد بود:

**مرحله 1:** در طول رشته‌ی ورودی w، از چپ به راست حرکت می‌کند و بررسی می‌نماید که آیا w عضوی از زبان توصیف شده با عبارت منظم $a^*b^*c^*$ است یا خیر. اگر نباشد، آنگاه رشته پذیرش نمی‌شود. در غیر این صورت به مرحله‌ی 2 می‌رود.

**مرحله 2:** به سمت چپ‌ترین نماد رشته‌ی w بر می‌گردد. به مرحله‌ی 3 می‌رود.

**مرحله 3:** در این مرحله، ماشین تورینگ موارد زیر را انجام می‌دهد:

- سمت چپ‌ترین a را با نماد خالی □ جایگزین می‌کند.
- به سمت چپ‌ترین b حرکت می‌نماید.
- هر لحظه بین bها و cها زیگ‌زاگ حرکت می‌کند، و سمت چپ‌ترین b را با نماد $، و سمت راست‌ترین c را با نماد خالی □، جایگزین می‌نماید. اگر برای b ای، هیچ c ای نمانده باشد، ماشین تورینگ رشته‌ی ورودی را رد می‌کند.
- حرکت زیگ‌زاگی را تا زمانی که هیچ b ای نمانده باشد، ادامه می‌دهد. سپس به مرحله‌ی 4 می‌رود.

مشاهده کنید که با این سه مرحله، رشته‌ی $a^m b^n c^k$ به رشته‌ی $a^{m-1}\$^n c^{k-n}$ انتقال می‌یابد.

**مرحله 4:** در این مرحله، ماشین تورینگ موارد زیر را انجام می‌دهد:

- هر $ را با b جایگزین می‌کند.
- به سمت چپ‌ترین a حرکت می‌نماید.

پس در این چهار مرحله، رشته‌ی $a^{m-1}\$^n c^{k-n}$ به رشته‌ی $a^{m-1}b^n c^{k-n}$ انتقال یافته است. ملاحظه می‌شود که رشته‌ی $a^m b^n c^k$ در زبان قرار دارد، اگر و تنها اگر رشته‌ی $a^{m-1}b^n c^{k-n}$ در زبان قرار داشته باشد. بنابراین، ماشین تورینگ مراحل 3 و 4 را، تا زمانی که هیچ a ای نمانده باشد، ادامه می‌دهد. در آن هنگام، بررسی می‌کند که آیا هیچ c ای مانده است یا خیر، اگر مانده باشد، رشته‌ی ورودی رد می‌شود، در غیر این صورت، رشته‌ی ورودی پذیرش می‌گردد.

## 8-3: ماشین‌های تورینگ چند نواری

در بخش پیش دو ماشین تورینگ دیدیم که متقارن‌ها را می‌پذیرد: ماشین تورینگ اول یک نوار دارد، در حالی که دومی دو نواری است.

توجه کردید که ماشین تورینگ دو نواری از ماشین تورینگ یک نواری ساده‌تر بدست می‌آید. این موضوع به این سوال برمی‌گردد که آیا ماشین‌های تورینگ چند نواری از همتای یک نواری‌شان قوی‌تر است؛ پاسخ منفی است.

**قضیه 8-3-1:** فرض کنیم $k \geq ۱$ عددی صحیح باشد. هر ماشین تورینگ k نواری می‌تواند به یک ماشین یک نواری معادل آن، تبدیل شود.

**اثبات:** اثبات را برای حالتی که $n = ۲$ انجام می‌دهیم. فرض کنیم:

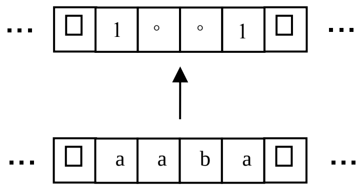
$M = (\Sigma, \Gamma, Q, \delta, q, q_{accept}, q_{reject})$ یک ماشین تورینگ دو نواری باشد. هدف این است که M را به یک ماشین یک نواری N که معادل آن است، تبدیل کنیم. یعنی، N باید به ازای هر رشته‌های $w \in \Sigma^*$ دارای خواص زیر باشد:

- M رشته w را پذیرش کند، اگر و تنها اگر N این رشته را بپذیرد،
- M رشته w را رد کند، اگر و تنها اگر N این رد نماید،
- M بر روی رشته ورودی w، متوقف نشود، اگر و تنها اگر N بر روی آن متوقف نشود، الفبای نوار ماشین تورینگ یک نواری N به صورت زیر باشد:

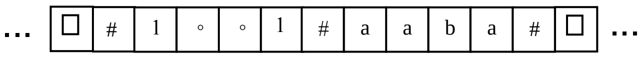
$$\Gamma \bigcup \{x^{\bullet} \mid x \in \Gamma\} \bigcup \{\#\}$$

یعنی الفبای نواری $\Gamma$ از M را به ازای هر $x \in \Gamma$ ، برداشته و نماد $x^{\bullet}$ را به آن می‌افزاییم. همچنین نماد خاص $ را نیز اضافه می‌کنیم.

ماشین تورینگ N به این ترتیب ساخته می‌شود که هر شکل‌گیری ماشین تورینگ دو نواری M، مثلاً:



با شکل گیری ماشین تورینگ یک نواری N، به صورت زیر متناظر باشد:



پس محتوی ماشین دو نواری M به رمزهای ماشین یک نواری N تبدیل شده است. نمادهای نقطه‌ها، مکان‌های دو هد نوار M را نشان می‌دهند، در حالی که سه رخداد نماد خاص # برای علامت زدن مرزهای رشته‌ها بر روی دو نوار ماشین M، بکار رفته‌اند.

ماشین تورینگ N یک گام از محاسبات M را به صورت زیر شبیه سازی می‌کند:

- در سرتاسر این شبیه سازی این گام، N، حالت جاری M را بخاطر دارد.
- در شروع شبیه سازی، هد نوار ماشین N در سمت چپ‌ترین نماد # قرار می‌گیرد.
- N در طول رشته به راست حرکت می‌کند، تا زمانی که اولین نماد نقطه‌ای را پیدا کند. (این نماد نشان دهنده مکان هد بر روی نوار اول M می‌باشد.) N این اولین نماد نقطه‌ای را بخاطر می‌سپارد و حرکت خود را به سمت راست ادامه می‌دهد، تا زمانی که دومین نماد نقطه‌ای را پیدا کند. (این نماد نشان دهنده مکان هد بر روی نوار دوم M می‌باشد.) بار دیگر، N این دومین نماد نقطه‌ای را بخاطر می‌سپارد.
- در این لحظه، N هنوز در دومین نماد نقطه‌ای است. N این قسمت نوار را به روز می‌کند، با تغییری که M بر روی نوار دوم خود انجام خواهد داد. (این تغییر توسط تابع انتقال ماشین M انجام می‌شود و به حالت جاری M و دو نمادی که M آنها را بر روی دو نوارش خوانده است، بستگی دارد.)
- N به چپ می‌رود تا زمانی که اولین نماد نقطه‌ای را پیدا کند. آنگاه، این بخش از نوار را به روز می‌کند، با تغییری که M بر روی اولین نوارش انجام می‌دهد.
- در دو گام قبلی که در آنها نوار به روز شده است، ممکن است انتقال یک بخش از نوار لازم باشد.

- سرانجام N حالت جدید M را بخاطر می‌سپارد و به سمت چپ‌ترین نماد # برمی‌گردد. روشن است که ماشین تورینگ N با معرفی حالت‌های مناسب می‌تواند ساخته شود.

## 8-3-2: نظریه چرچ- تورینگ

همه ما با مفهوم واقعی الگوریتم آشنایی پیدا کرده‌ایم. این مفهوم را شاید بشود چنین بیان کرد،"یک الگوریتم روشی است شامل گام‌های محاسباتی که می‌توانند در بخشی از متن مشخص گردند". برای مثال هر "فرآیند محاسباتی" که بتواند با یک برنامه Java معین شود، می‌تواند به عنوان یک الگوریتم در نظر گرفته شود. مشابهاً ماشین تورینگ یک "فرآیند محاسباتی" را مشخص می‌کند و بنابراین باید به عنوان یک الگوریتم در نظر گرفته شود. این امر موجب طرح این سوال می‌گردد که آیا می‌توان یک تعریف ریاضی از یک الگوریتم ارایه داد؟ بیان کردیم که هر برنامه Java یک الگوریتم را نشان می‌دهد و همچنین گفتیم که هر ماشین تورینگ نیز یک الگوریتم را نمایش می‌دهد. آیا این دو مفهوم از الگوریتم با هم معادل‌اند؟ جواب آری است. در واقع قضیه زیر بیان می‌کند که بسیاری از مفاهیم مختلف "فرآیند محاسباتی" با هم معادل‌اند.

**قضیه 8-3-3:** مدل‌های محاسباتی زیر معادل‌اند، یعنی هر یک از آنها می‌تواند به یکی دیگر تبدیل شود:

1. ماشین‌های تورینگ یک نواری.
2. ماشین‌های تورینگ k نواری، به ازای هر $k \geq ۱$ .
3. ماشین‌های تورینگ نامعین.
4. برنامه‌های Java.
5. برنامه‌های ++C.
6. برنامه‌های Lisp.

به بیان دیگر، اگر مفهوم الگوریتم با استفاده از هر یک از مدل‌های این قضیه تعریف شود، آنگاه تفاوتی نمی‌کند که کدام مدل را برداریم. هم این مدل‌ها یک مفهوم یکسان از الگوریتم ارایه می‌نمایند.

مسأله تعریف مفهوم یک الگوریتم به ریاضیدان مشهور، دیوید هیلبرت (David Hilbert) بر می‌گردد. در هشتم ماه آگوست 1900، در دومین کنگره بین المللی ریاضی که در پاریس برگزار گردید، هیلبرت فهرستی از مسائلی مطرح کرد که معتقد بود برای توسعه ریاضی قطعی هستند و دهمین مسأله‌ی هیلبرت عبارتست از: آیا یک روند متناهی وجود دارد که بتواند درباره این موضوع تصمیم بگیرد که آیا هر چند جمله‌ای مفروض با ضرایب صحیح دارای ریشه‌های صحیح است؟ می‌توانیم این مطلب را چنین بیان کنیم که:

هیلبرت سوال کرد که آیا یک الگوریتم وجود دارد که بتواند درباره موضوع زیر تصمیم بگیرد:

آیا چند جمله‌ای مفروض

$$12x^3y^7z^5 + 7x^2y^4z - x^4 + y^3z^7 - z^3 + 10 = 0$$

دارای ریشه‌های صحیح است یا خیر؟ (ضرایب اعداد صحیح‌اند).

در 1970، ماتیا سویچ (Matia Sevich) ثابت کرد که چنین الگوریتمی وجود ندارد. البته برای اثبات این ادعا، ابتدا باید درباره این که"یک الگوریتم چیست" به توافق برسیم. در آغاز قرن بیستم، ریاضیدانان چندین تعریف ارایه نمودند، نظیر ماشین تورینگ (1963) و حساب $\lambda -$ $(\lambda - \text{Calculus})$ و ثابت کردند که همه آنها معادل‌اند. بعدها پس از اختراع زبان‌های برنامه‌نویسی، نشان داده شد که این مفاهیم قدیمی از الگوریتم، با مفاهیم الگوریتمی که بر پایه برنامه‌نویسی به زبان C، Java و Lisp، پاسکال و غیره. تنظیم نشده است، معادل می‌باشند.

به بیان دیگر، همه تلاش‌ها بر این قرار گرفت که یک تعریف دقیق از مفهوم الگوریتم ارایه شود که به درک یکسان منجر گردد. به این دلیل، دانشمندان کامپیوتر این روزها بر نظریه چرچ- تورینگ توافق کرده‌اند.

## نظریه چرچ- تورینگ

هر فرآیند محاسباتی که مستقیماً یک الگوریتم در نظر گرفته شود، می‌تواند به ماشین تورینگ تبدیل گردد. به عبارت دیگر، این موضوع بیان می‌کند که یک الگوریتم را به عنوان یک ماشین تورینگ تعریف کنیم. پس در این لحظه، می‌توانیم فکر کنیم که نظریه چرچ- تورینگ اثبات پذیر است.

به بیان دیگر چه چیزی باید انجام گیرد تا نشان دهد این قضیه اثبات شدنی نیست؟