تکلیف ۱۰

مدل‌های دیگر ماشین تورینگ

Other Models of Turing Machines

۱. فرض کنید یک محدودیت بر روی ماشین تورینگ استاندارد قرار دهیم که همیشه باید نمادی که روی نوار می‌نویسد با نمادی که خوانده است، متفاوت باشد، یعنی

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

آیا این محدودیت قدرت این ماشین را کاهش می‌دهد؟ چرا؟

۲. یک ماشین تورینگ چندهده (multihead) می‌تواند به صورت ماشین تورینگی با یک نوار واحد و یک واحد کنترل اما با هدهای خواندن - نوشتن چندگانه‌ی مستقل تصور شود. یک تعریف رسمی برای ماشین تورینگ چندهده ارائه دهید و سپس نشان دهید که چگونه چنین ماشینی می‌تواند با یک ماشین تورینگ استاندارد شبیه‌سازی شود.

۳. **آتوماتون صفی** (queue automaton) آتوماتونی است که حافظه‌ی موقت آن یک صف باشد. فرض کنید که این ماشین یک ماشین برخط (online) باشد، یعنی فایل ورودی ندارد و رشته‌ای که باید پردازش شود، پیش از شروع محاسبه در صف قرار داده می‌شود. یک تعریف رسمی برای این آتوماتون ارائه دهید و قدرت آن را در مقایسه با ماشین تورینگ بررسی کنید.

۴. یک برنامه برای ماشین تورینگ غیرقطعی بنویسید که زبان زیر را می‌پذیرد:

$$L = \{ww : w \in \{a, b\}^+\}$$

۵. یک ماشین تورینگ غیرقطعی طراحی کنید که زبان زیر را می‌پذیرد:

$$L = \{a^n : n \text{ is not a prime number}\}$$

۶. یک آتوماتون دو پشته‌ای، یک آتوماتون پشته‌ای با دو پشته‌ی مستقل است. برای تعریف این آتوماتون تعریف NPDA را به صورت زیر تغییر می‌دهیم:

$$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \times \Gamma \rightarrow \text{finite subsets of } Q \times \Gamma^* \times \Gamma^*$$

یک حرکت وابسته به بالای دو پشته است و نتیجه‌های جدید در این دو پشته قرار می‌گیرند. نشان دهید که کلاس آتوماتای دو پشته‌ای هم‌ارز با کلاس ماشین‌های تورینگ است.

۷. طرح اولیه‌ای برای ماشین تورینگی که مجموعه‌ی $\{۰, ۱\}^+$ را به ترتیب سره (proper order) می‌شمارد، ارائه دهید.

۸. فرض کنید که $S_۱$ و $S_۲$ دو مجموعه‌ی شمارا باشند. نشان دهید که $S_۱ \cup S_۲$ و $S_۱ \times S_۲$ مجموعه‌های شمارا هستند.

۹. چگونه می‌توان برای پذیرش زبان $\{a^{n!} : n \geq ۰\}$ یک LBA ارائه کرد که به شیار اضافی نیاز نداشته باشد؟

۱۰. نشان دهید که برای هر زبان مستقل از متن، یک PDAی پذیرنده وجود دارد که تعداد نمادهای آن در پشته با بیش از یک واحد از طول ورودی تجاوز نمی‌کند.

۱) خیر، برای هر نماد $a \in T$ یک شبه‌نماد تعریف می‌کنیم (مانند A). هرگاه لازم داشتیم a خوانده و نوشته شود (تغییر نکند)، ابتدا A را می‌نویسیم و سپس به سلول مورد نظر برمی‌گردیم و A را با a جایگزین می‌نماییم.

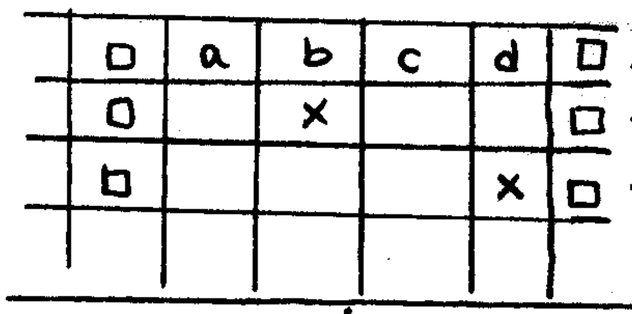
---

۲) تعریف صوری:

$$\delta: Q \times T_T \longrightarrow Q \times T_T \times \{L, R\}^m \qquad T_T = \underbrace{T \times T \times \cdots \times T}_{m's}$$

m: تعداد هدهای خواندن و نوشتن

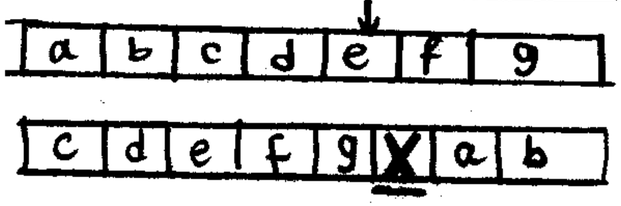
← وقتی دو هد خواندن و نوشتن بر روی یک سلول باشند چه اتفاقی می‌افتد؟ تعریف صوری باید برای رفع تداخل‌های ممکن امکاناتی را فراهم کند.



برای شبیه‌سازی ماشین اصلی (OM) با ماشین تورینگ استاندارد، (SM)، فرض می‌کنیم (SM) دارای m+1 نوار باشد.
بر روی یک نوار محتوای نوار OM و بر روی m نوار دیگر مکان هدهای نوار OM نوشته می‌شود.
SM هر حرکت OM را با اسکن کردن و بهنگام کردن ناحیه‌ی فعال آن شبیه‌سازی می‌نماید.

---

۳) آتوماتون صفی معادل با ماشین تورینگ استاندارد است، پس صف یک حافظه‌ی قدرتمندتر نسبت به پشته است.
برای شبیه‌سازی یک TM استاندارد با ماشین صفی می‌توانیم مثلاً ست راست OM را در جلوی صف و ست چپ آن را در پشت قرار دهد.
حرکت به راست به سادگی حذف نماد جلوی صف و قرار دادن آن در انتهاست.
حرکت به چپ معادل با چندبار چرخش است تا هم چیز در محل مناسب خود قرار گیرد.



استفاده از نمادهای کمکی Y و Z برای نمایش مرزها مفید است. برای مثال برای شبیه‌سازی $\delta(q_i, c) = (q_j, z, L)$ گام‌های زیر را داریم:

۱) حذف c از جلو و اضافه کردن Yz به انتها ۲) چرخش محتوای صف برای به دست آوردن: bzYdefgXa ۳) اضافه کردن z به انتها، چرخش، حذف Y و z در صورت واقع شدن در جلوی صف

---

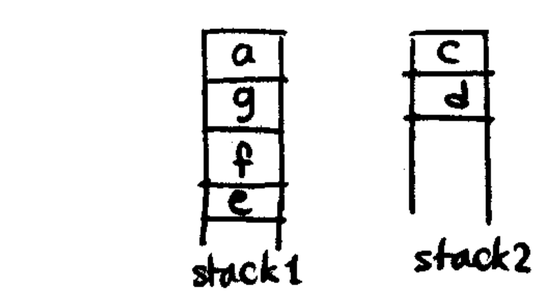
۴) از سمت چپ ورودی شروع می‌کنیم. نماد خوانده شده را با قرار دادن ماشین در حالت مناسب به خاطر می‌آوریم، سپس آن را با X جایگزین می‌نماییم.
- هد خواندن ـ نوشتن را به راست حرکت می‌دهیم، توقف (به طور غیرقطعی) در وسط ورودی.
- نماد واقع در اینجا را با نماد به خاطر سپرده شده مقایسه می‌کنیم. در صورت تطابق، Y را در سلول می‌نویسیم. اگر تطابق نداشت، رشته رد می‌شود.
- با علامتگذاری وسط رشته با Y، اکنون می‌توانیم به طور قطعی ادامه بدهیم. (حرکت به راست و چپ یک در میان و مقایسه‌ی نمادها)

---

۵) به طور غیرقطعی مقداری برای n انتخاب می‌کنیم. تعیین می‌کنیم که آیا طول ورودی مضربی از n است یا خیر، اگر بود پذیرش صورت می‌گیرد.
اگر $a^n \in L$ باشد در این صورت یک n وجود دارد که این برای آن درست عمل می‌کند. $2 \le n \le |w|$

---

۶) stack1 حاوی نماد زیر هد خواندن ـ نوشتن ماشین OM و هر چیز سمت چپ آن است.
stack2 حاوی همه‌ی اطلاعات سمت راست هد خواندن ـ نوشتن است.
حرکت‌های چپ و راست به سادگی شبیه‌سازی می‌شوند.
برای مثال $\delta(q_i, a) = (q_j, b, L)$
به صورت زیر شبیه‌سازی می‌شود: - pop کردن a از stack1
- قرار دادن یک b بر روی stack2.



---

۷) طرح الگوریتم: ۱ با یک کپی از رشته‌ی قبلی شروع می‌کنیم. $\{0,1\}^+$
۲ سمت راست‌ترین 0 را می‌یابیم. آن را به 1 تبدیل می‌کنیم. سپس همه‌ی 1های سمت راست آن را به 0 تبدیل می‌کنیم.
۳ اگر هیچ 0 ای وجود نداشت، همه‌ی 1ها را به 0 تبدیل می‌کنیم و یک 1 به سمت چپ می‌افزاییم.
۴ از گام ۱ شروع می‌کنیم و ادامه می‌دهیم.

---

۸) $S_1 = \{s_1, s_2, \cdots\}$, $S_2 = \{t_1, t_2, \cdots\}$

$S_1 \cup S_2 = \{s_1, t_1, s_2, t_2, \cdots\}$ → countable

$S_1 \times S_2$: مشابه شمارش اعداد گویا → countable

---

۹) ابتدا ورودی را به دو تقسیم می‌کنیم و نتیجه را به یک بخش از نوار انتقال می‌دهیم. این فضای آزاد در ابتدا توسط ورودی اشغال شده است.
از این فضا می‌توان برای ذخیره کردن divisor های متوالی استفاده کرد.

---

۱۰) (فصل ۶ سه گرامر دو استانداردند): برای هر گرامر مستقل از متن می‌توان گرامر دو استانداردی معادل را پیدا کرد.
پس از ساختار قضیه‌ی ۷ـ۱ برای ایجاد PDA استفاده نمود.
PDA به دست آمده از این ساختار در هر حرکت یک نماد ورودی را مصرف می‌کند و هرگز محتوای پشته را بیش از یک نماد در هر مرتبه افزایش نمی‌دهد.