به نام خدا

نظریه زبان‌ها و ماشین‌ها – بهار ۱۴۰۲

تمرین شماره ۱۱

دستیار آموزشی این مجموعه: سامان اسلامی نظری

Saman.Eslami78@gmail.com

تاریخ تحویل: ۱۴ خرداد (صفحه درس)

۱. فرض می‌کنیم که زبان EQ_{CFG} تصمیم‌پذیر است؛ زبان ALL_{CFG} را به آن کاهش می‌دهیم:

ماشین تورینگ M را برای زبان ALL_{CFG}، به صورت زیر می‌سازیم:

M = "به ازای ورودی $\langle G \rangle$:

۱. یک گرامر H که زبان آن $\Sigma *$ را می‌سازیم.

۲. ماشین تصمیم‌گیرنده EQ_{CFG} را به ازای ورودی $\langle G, H \rangle$ اجرا می‌کنیم.

۳. در صورتی که آن را قبول کرد ما نیز ورودی را قبول می‌کنیم. در غیر این صورت ورودی را رد می‌کنیم."

۲. فرض می‌کنیم مسئله مذکور توسط ماشین تورینگ H تصمیم‌پذیر است. مسئله $\mathrm{A_{TM}}$ را به این مسئله، به روش زیر، کاهش می‌دهیم:

S = "به ازای ورودی $\langle M, w \rangle$:

۱. از M برای ساختن ماشین تورینگ دو نواره به صورت زیر استفاده می‌کنیم:

T = "به ازای ورودی x:

۱. ماشین M را روی نوار اول شبیه‌سازی می‌کنیم.

۲. اگر شبیه‌سازی موفق بود، روی نوار دوم یک کاراکتر می‌نویسیم."

۳. ماشین H را روی ورودی $\langle \mathrm{T}, \mathrm{w} \rangle$ اجرا می‌کنیم. اگر آن را قبول کرد، ورودی را می‌پذیریم و در غیر این صورت آن را رد می‌کنیم."

۳. بدون از دست دادن کلیت مسئله، فرض می‌کنیم که الفبای این زبان تنها شامل 1 می‌شود. بنابراین مسئله PCP به صورت زیر خواهد بود:

$$P = \left\{ \left[\frac{1^{\mathrm{a}_1}}{1^{\mathrm{b}_1}} \right], \left[\frac{1^{\mathrm{a}_2}}{1^{\mathrm{b}_3}} \right], \ldots, \left[\frac{1^{\mathrm{a}_\mathrm{n}}}{1^{\mathrm{b}_\mathrm{n}}} \right] \right\}$$

ماشین تورینگ زیر را برای تصمیم‌گیری این زبان می‌سازیم:

M = "برای ورودی $\langle \mathrm{P} \rangle$:

۱. اگر به ازای یک i-ای، $\mathrm{a}_\mathrm{i} = \mathrm{b}_\mathrm{i}$ وجود داشت، ورودی را قبول کن.

۲. اگر به ازای یک i که $\mathrm{a}_\mathrm{i} > \mathrm{b}_\mathrm{i}$، یک j وجود داشت که $\mathrm{a}_\mathrm{j} < \mathrm{b}_\mathrm{j}$ بود آنگاه ورودی را قبول کن. در غیر این صورت ورودی را رد کن."

دلیل بخش اول واضح است. برای بخش دوم، برای حل مسئله، تنها کافیست با استفاده از کسر i تعداد یک‌های صورت کسرها را به اندازه ک.م.م $\mathrm{a}_\mathrm{i} - \mathrm{b}_\mathrm{i}$ و $\mathrm{a}_\mathrm{j} - \mathrm{b}_\mathrm{j}$ بیش‌تر از تعداد یک‌های مخرج کسر کرده و سپس این اختلاف را با استفاده از کسر j-ام جبران می‌کنیم. از آنجا که در هر حالت این ماشین تورینگ توقف می‌کند، زبان داده شده تصمیم‌پذیر است.

۴. اثبات می‌کنیم که $\mathrm{A}_{\mathrm{tm}} \leq_{\mathrm{m}} T$. برای اینکار نگاشت $\langle \mathrm{M}, \mathrm{w} \rangle$ به $\langle \mathrm{M}' \rangle$ را ارائه می‌دهیم:

ماشین M': "به ازای ورودی x:

۱. در صورتی که $x \neq ab$ یا $x \neq ba$ باشد، به استیت reject می‌رویم.
۲. در صورتی که ورودی برابر ab باشد، آن را قبول می‌کنیم.
۳. در صورتی که ورودی برابر ba باشد، ابتدا ماشین تورینگ M را به ازای ورودی w اجرا کرده، و اگر آن را قبول ما نیز به استیت accepting می‌رویم. در غیر این صورت به استیت reject می‌رویم."

واضح است که اگر $\langle \mathrm{M}, \mathrm{w} \rangle \in \mathrm{A}_{\mathrm{TM}}$ آنگاه M ورودی w را قبول کرده و $L(\mathrm{M}') = \{\mathrm{ab}, \mathrm{ba}\}$ و در این صورت T نیز M' را قبول می‌کند. اما اگر $\langle \mathrm{M}, \mathrm{w} \rangle \notin \mathrm{A}_{\mathrm{TM}}$ آنگاه $L(\mathrm{M}') = \{\mathrm{ab}\}$ و T نیز M' را قبول نمی‌کند. بنابراین، این نگاشت صحیح می‌باشد

۵. مسئله PCP را به این زبان کاهش می‌دهیم. یک نمونه از این مسئله به صورت $P = \left\{\left[\frac{\mathrm{t_1}}{\mathrm{b_1}}\right], \left[\frac{\mathrm{t_2}}{\mathrm{b_2}}\right], \dots, \left[\frac{\mathrm{t_k}}{\mathrm{b_k}}\right]\right\}$ در نظر می‌گیریم. گرامر زیر را متناظر با این نمونه از PCP می‌سازیم.

$$S \to \mathrm{S_1} \mid \mathrm{S_2}$$
$$\mathrm{S_1} \to \mathrm{t_1 S_1 a_1} \mid \dots \mid \mathrm{t_k S_1 a_1} \mid \mathrm{t_1 a_1} \mid \dots \mid \mathrm{t_k a_k}$$
$$\mathrm{S_2} \to \mathrm{b_1 S_1 a_1} \mid \dots \mid \mathrm{b_k S_1 a_1} \mid \mathrm{b_1 a_1} \mid \dots \mid \mathrm{b_k a_k}$$

در اینجا کاراکترهای $\mathrm{a_1, a_2, \dots, a_k}$ کاراکترهای اضافی هستن که در مسئله PCP-مان موجود نیستند. حال اگر این گرامر مبهم باشد، یک رشته w را به دو صورت از قوانین بالا می‌توان بدست آورد. بدون از دست دادن کلیت مسئله، فرض می‌کنیم که با قانون $S \to \mathrm{S_1}$ شروع کرده و به دو صورت رشته مورد نظر را می‌سازیم. در صورتی که رشته موردنظر توسط گرامر پذیرفته شود، انتهای آن شامل کاراکترهای جدید خواهد بود؛ از آنجا که تنها به یک صورت از طریق $\mathrm{S_1}$ می‌توان این کاراکترهای جدید را ساخت، بنابراین تنها یک راه از طریق قانون $S \to \mathrm{S_1}$ برای ساخت رشته w خواهیم داشت. پس اگر گرارم مبهم باشد، یک درخت اشتقاق از قانون $S \to \mathrm{S_1}$ و درخت دیگر از قانون $S \to \mathrm{S_2}$ استفاده خواهد کرد. حال در صورتی که این گرامر مبهم باشد یک رشته مانند w وجود خواهد داشت که یک بار از راه قانون اول و بار دیگر از راه قانون دوم پذیرفته می‌شود. در این رشته، زیررشته سمت چپ آن که شامل کاراکترهای PCP است نیز با یکدیگر یکسان خواهند بود و این زیررشته نشان‌دهنده یک راه حل برای مسئله PCP است.

در نهایت با ساخت گرامری که گفته شد، می‌توان جواب مسئله PCP داده شده را بدست آورد. اگر $AMBIG_{\mathrm{CFG}}$ بگوید که گرامر ورودی مبهم است، مسئله PCP قابل حل و در صورتی که بگوید مبهم نیست، مسئله PCP غیرقابل حل خواهد بود. بنابراین مسئله PCP که یک مسئله undecidable است به این زبان کاهش داده شد. پس این زبان نیز undecidable است.

۶. فرض می‌کنیم مسئله $\mathrm{A_{TM}}$ توسط ماشین تورینگ $\mathrm{H_1}$ قابل شناسایی باشد. ماشین تورینگ $\mathrm{H_2}$ را می‌سازیم که به ازای ورودی $\langle \mathrm{n} \rangle$ مسئله $3x + 1$ را اجرا می‌کند؛ واضح است که اگر به ازای n دنباله در جایی به ۱ برسد متوقف شده و ورودی را قبول می‌کند؛ در غیر این صورت تا ابد در چرخه گیر خواهد کرد. ماشین تورینگ $\mathrm{H_3}$ را می‌سازیم که به ازای ورودی $\langle \mathrm{H_2, n} \rangle$، ماشین $\mathrm{H_1}$ را اجرا کرده و اگر آن را قبول کرد، ورودی را قبول می‌کند و در غیر این صورت آن را رد می‌کند (در اصل با استفاده از ماشین تصمیم‌گیرنده $\mathrm{A_{TM}}$، مسئله $3x + 1$ را به ازای یک ورودی خاص حل کردیم). اکنون ماشین $\mathrm{H_4}$ را تعریف می‌کنیم که در آن به ازای هر ورودی، ماشین $\mathrm{H_3}$ را با n از ۱ تا بی‌نهایت اجرا می‌کند؛ در صورتی که یک عددی وجود داشته باشد که به ازای آن، دنباله به یک ختم نشود، ماشین از کار ایستاده و به یک accepting state می‌رود. در غیر این صورت ماشین هیچگاه از کار نخواهد ایستاد. در نهایت ماشین $\mathrm{H_5}$ را تعریف می‌کنیم که به ازای هر ورودی، ماشین $\mathrm{H_4}$ را با همان ورودی خودش (دقت کنید که برای ماشین $\mathrm{H_4}$ مقدار ورودی اهمیتی ندارد) به $\mathrm{H_1}$ می‌دهیم؛ اگر $\mathrm{H_1}$ آن را پذیرفت، می‌فهمیم که عددی وجود دارد که به ازای آن دنباله $3x + 1$ به ۱ ختم نمی‌شود؛ اگر $\mathrm{H_1}$ آن را نپذیرفت، یعنی دنباله $3x + 1$ به ازای تمام اعداد طبیعی به ۱ ختم می‌شود. ماشین $\mathrm{H_5}$ جواب مسئله است.

۷. زبان E_{TM} نشان‌گر ماشین‌ها TM-ای است که زبانشان تهی است. می‌دانیم این زبان تصمیم‌پذیر نیست. مسئله مذکور در صورت سوال را به صورت زبان زیر نشان می‌دهیم:

$$USELESS_{TM} = \{\langle M, q\rangle | q \text{ is a useless state in } M\}$$

فرض می‌کنیم که $USELESS_{TM}$ تصمیم‌پذیر بوده و توسط ماشین H حل می‌شود. زبان E_{TM} را به این زبان کاهش می‌دهیم:

به ازای هر استیت پذیرنده (accepting) ماشین ورودی M به صورت $q_{accepting}$، ماشین H را با ورودی $\langle M, q_{accepting}\rangle$ اجرا می‌کنیم. در صورتی که H مشخص کند تمام $q_{accepting}$-ها در ماشین M بی‌فایده هستند، آنگاه متوجه می‌شویم که $L(M) = \emptyset$. بنابراین بدین صورت توانستیم مسئله E_{TM} را حل کنیم. اما می‌دانیم E_{TM} تصمیم‌پذیر نیست؛ پس فرض خلف باطل بوده و $USELESS_{TM}$ نیز تصمیم‌پذیر نمی‌باشد.

۸. نشان می‌دهیم که اگر این تابع قابل محاسبه باشد، آنگاه $\mathrm{A_{TM}}$ نیز decidable خواهد بود. فرض می‌کنیم تابع BB توسط ماشین F محاسبه می‌شود. ماشین S را برای $\mathrm{A_{TM}}$ به صورت زیر می‌سازیم:

ماشین S = "به ازای ورودی $\langle \mathrm{M}, \mathrm{w}\rangle$:

۱. ماشین $\mathrm{M_W}$ را روی الفبای نوار Γ به صورت زیر می‌سازیم:

$\mathrm{M_W}$ = "به ازای هر ورودی:

۱. ماشین M را روی ورودی w شبیه‌سازی کن. در این حین تعداد استیت‌هایی که در حین شبیه‌سازی رد می‌شوند را ذخیره کن.

۲. اگر M محاسباتش تمام شد (halt کرد)، به تعداد استیت‌هایی که در مرحله قبل برای شبیه‌سازی شمردیم، عدد یک روی نوار بنویس."

۲. با استفاده از ماشین F مقدار $b = BB(\mathrm{k})$ را بدست آورد. k در اینجا تعداد استیت‌های $\mathrm{M_W}$ است.

۳. M را به ازای w به اندازه b قدم اجرا کن.

۴. در صورتی که M با این تعداد قدم اجرا ورودی را قبول کرد، ما نیز به accepting استیت می‌رویم. در غیر اینصورت به یک rejecting استیت می‌رویم."

در صورتی که M ورودی w را قبول کند، آنگاه M_W به تعداد قدم‌های اجرای M روی نوار، یک می‌نویسد. علاوه بر آن، طبق تعریف BB می‌دانیم که مقدار b از تعداد یک‌هایی که روی نوار نوشتیم بیش‌تر یا برابر آن است (چون از بین تمام ماشین تورینگ‌هایی که اندازه M_W استیت دارند، b نشان‌دهنده بیش‌ترین تعداد یک‌های روی نوار این ماشین‌هاست؛ تعداد یک‌های

روی نوار در انتهای محاسبات نیز قطعا از تعداد مراحل اجرای یک ماشین تورینگ بیش‌تر نخواهد بود). بنابراین S ماشین M را به اندازه کافی اجرا می‌کند تا متوجه شود که ورودی را قبول می‌کند و خودش نیز ورودی را قبول کند. اگر M ورودی را قبول نکند، S نیز هیچگاه قبول کردن M را ندیده و ورودی را رد می‌کند.