autor dokumetu: Rafał Stefański

Zwiększenie rzędu stosu zwiększa klasę języków rozpoznawanych przez automat ze stosem

W pierwszej części wykładu pokażemy, że dla każdego $n > 1$ istnieje język L rozpoznawany przez deterministyczny automat ze stosem rzędu n bez operacji `collapse`, który nie jest rozpoznawany przez żaden niedeterministyczny automat ze stosem rzędu $n-1$ nawet z `collapse`.

Przykładem takiego języka będzie L – język poprawnych operacji stosowych nad następującym alfabetem:

$$\Sigma = \{\mathtt{pop}_1, \mathtt{pop}_2, \ldots, \mathtt{pop}_n, \mathtt{push}_1(\mathtt{a}), \mathtt{push}_2(\mathtt{b}), \mathtt{push}_2, \mathtt{push}_3, \ldots, \mathtt{push}_n, \mathtt{top} = \mathtt{a}, \mathtt{top} = \mathtt{b}\}$$

Automat ze stosem rzędu n bez problemu rozpozna taki język – może po prostu symulować wszystkie te operacje na swoim stosie. Przypuśćmy, że istnieje $\mathcal{B}$ rzędu $n-1$ rozpoznający L. Pokażmy wówczas, że dla dowolnego $\mathcal{A}$ ze stosem rzędu n istnieje $\mathcal{A}'$ ze stosem rzędu $n-1$, który rozpoznaje ten sam język co $\mathcal{A}$. Stany $\mathcal{A}'$ to $Q_\mathcal{A} \times Q_\mathcal{B}$. $\mathcal{A}'$ symuluje $\mathcal{A}$, a kiedy chce wykonać operację stosową „przekazuje” odpowiednią literę automatowi $\mathcal{B}$. Słowo jest akceptowane, gdy zarówno stan z $\mathcal{A}$ i $\mathcal{B}$ w którym znajduje się $\mathcal{A}'$ na końcu słowa są akceptujące. Tak zdefiniowany automat akceptuje ten sam język co $\mathcal{A}$. Zauważmy, że dla ustalonego n wielkość $\mathcal{B}$ jest stała, czyli rozmiar $\mathcal{A}'$ jest wielomianowy od rozmiaru $\mathcal{A}$. W związku z czym problem niepustości języka rozpoznawanego przez dowolny automat ze stosem rzędu n redukuje się wielomianowo do problemu niepustości języka rozpoznawanego przez automat o stosie rzędu $n-2$. Poprzednio udowodniliśmy, że pierwszy problem jest $(n-1)$-EXPTIME-trudny, a drugi problem jest $(n-2)$-EXPTIME-trudny. Prowadzi to do sprzeczności, gdyż istnieją problemy, które należą do $(n-1)$-EXPTIME, a nie należą do $(n-2)$-EXPTIME. Wynika z tego, że $\mathcal{B}$ jednak nie istnieje, czyli, że L nie jest rozpoznawane przez żaden automat ze stosem rzędu $n-1$.

Jeśli przez PDA_n i $DPDA_n$ oznaczymy klasy języków rozpoznawanych przez odpowiednio niedeterministyczne i deterministyczne automaty ze stosem rzędu n to hierarchię automatów ze stosem można zapisać w sposób następujący:

$$\begin{array}{ccccccc}
PDA_0 & \subsetneq & PDA_1 & \subsetneq & PDA_2 & \subsetneq & \ldots \\
\parallel & & \cup\!\!\!\!\!\not\;\; & & \cup\!\!\!\!\!\not\;\; & & \ldots \\
DPDA_0 & \subsetneq & DPDA_1 & \subsetneq & DPDA_2 & \subsetneq & \ldots
\end{array}$$

Klasa języków rozpoznawanych przez automaty ze stosem rzędu n i k głowicami to dokładnie $(n-1)$-EXPTIME

Zdefiniujmy klasę automatów ze stosem rzędu n i k głowicami. Każdy taki automat ma k głowic, jeden, globalny stos rzędu n i jeden, globalny stan. Wejście takiego automatu jest wzbogacone o znaczniki początku i końca słowa ($\vdash$ i $\dashv$), a jego relacja przejścia wyglądają następująco:

$$\delta \subseteq \underbrace{Q}_{\text{obecny stan}} \times \underbrace{(\Sigma \cup \{\vdash, \dashv\})^k}_{\text{symbole pod głowicami}} \times \underbrace{\Gamma}_{\text{symbol na szczycie stosu}} \times \underbrace{Q}_{\text{nowy stan}} \times \underbrace{\{\leftarrow, \_, \rightarrow\}^k}_{\text{ruchy głowic}} \times \underbrace{\mathrm{Stack}_n^\Gamma}_{\text{operacja na stosie}}.$$

Wejście akceptowane jest w momencie, gdy automat przejdzie do specjalnego stanu.

Klasa języków rozpoznawanych przez takie automaty nie zmieni się, jeśli dodatkowo każda z głowic będzie mogła dowiedzieć się, które głowice stoją w tym samym miejscu co ona. Taką operacje można łatwo zasymulować używając kilku dodatkowych głowic. Znalezienie takiej symulacji pozostawiamy jako ćwiczenie dla czytelnika.

Pokażmy teraz, że dowolny język rozpoznawany przez tego typu automat należy do $(n-1)$-EXPTIME. Dla dowolnego automatu $\mathcal{A}$ z k głowicami i stosem rzędu n oraz dowolnego słowa w możemy łatwo skonstruować klasyczny automat ze stosem rzędu n (nazwijmy go $\mathcal{A}_w$), który akceptuje jakieś słowo (np. słowo puste) wtedy i tylko wtedy gdy $w \in L(\mathcal{A})$. Automat ten przechowuje pozycje głowic w stanie (może tak zrobić bo każde w ma pewną, skończoną długość), a operacje stosowe wykonuje normalnie na swoim stosie. Mając dane dowolne $\mathcal{A}$ możemy skonstruować $\mathcal{A}_w$ i w czasie $(n-1)$-EXPTIME sprawić, czy jest niepuste ($\mathcal{A}_w$ ma rozmiar wielomianowy od rozmiaru w). Zatem, istotnie, wszystkie języki rozpoznawane przez automaty ze stosem rzędu n i k głowicami należą do $(n-1)$–EXPTIME.

Pokażmy, w końcu, że każdy język z $(n-1)$-EXPTIME jest rozpoznawany przez pewien automat z k głowicami i stosem rzędu n. Zdefiniujmy najpierw język L_n – wszystkie opisy automatów ze stosem rzędu n, które akceptują przynajmniej jedno słowo. Opis automatu to opis wszystkich jego tranzycji, a przykładowy opis tranzycji wygląda tak:

$$\$\underbrace{\mathtt{0101001111}}_{\text{stan automatu}}\#\underbrace{\mathtt{a}}_{\text{litera na wejściu}}\#\underbrace{\mathtt{b}}_{\text{litera na syczycie stosu}}\#\underbrace{\mathtt{0000111010}}_{\text{nowy stan}}\#\underbrace{\mathtt{push}_5(\mathtt{a})}_{\text{operacje stosowe}}\$.$$

Zakładamy, że stan złożony z samych 0 jest początkowy, a stan złożony z samych 1 jest akceptujący. Język taki możemy rozpoznać używając do tego automatu ze stosem rzędu n, jedną głowicą i dodatkową logarytmiczną pamięcią. W logarytmicznej pamięci trzymamy aktualny stan, a na stosie zawartości stosu symulowanego automatu. W każdym kroku niedeterministycznie zgadujemy tranzycję – podjeżdżamy do jej opisu głowicą automatu i próbujemy wykonać. Jeśli się uda aktualizujemy stan automatu, jeśli nie kończymy bieg. Akceptujemy automat gdy dojdziemy do jego stanu akceptującego. (Obecny stan symulowanego automatu musimy trzymać w logarytmicznej pamięci, gdyż automaty z wejścia potencjalnie mogą mieć dowolnie wiele stanów). Zauważmy, że opisana na poprzednim wykładzie redukcja dowolnego języka z $(n-1)$-EXPTIME do języka L_n działa w logarytmicznej pamięci. Możemy ją w związku z tym „złożyć" z automatem rozpoznającym język L_n by otrzymać automat rzędu n z k stosami i dodatkową logarytmiczną pamięcią, który rozpoznaje dowolny język z $(n-1)$-EXPTIME. Ostatnia rzecz, którą musimy zauważyć to fakt, że korzystając z m dodatkowych głowic możemy zasymulować taśmę o rozmiarze $m \cdot \log x$, gdzie x oznacza długość wejścia – każda z głowic może znajdować się w jednym z x ustawień, a w związku z tym może symulować taśmę o długości $\log x$, gdyż głowicę na pozycji i potraktować jako opis kawałka taśmy zapisanego rozwinięciem dwójkowym liczby i. (Implementację operacji odczytu i zapisu na taśmę, ukrytą pod postacią pozycji głowic zostawiamy znowu jako ćwiczenie na czytelnika). Łącząc ten fakt z poprzednio uzyskaną konstrukcją, uzyskujemy automat z k głowicami i stosem rzędu n, który rozpoznaje dowolny język należący do $(n-1)$-EXPTIME. Pokazuje to, że klasa języków rozpoznawanych przez automaty ze stosem rzędu n i k głowic to dokładnie $(n-1)$-EXPTIME.