Programowanie

Egzamin zasadniczy — rozwiązania

14 czerwca 2002

Zadanie 1.

$\{\mathtt{X} = n \wedge n \geq 0\}$
$\{1 = h(n - \mathtt{X}) \wedge \mathtt{X} \geq 0\}$
`S = 1;`
$\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} \geq 0\}$
`while (X>0) (`
    $\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} \geq 0 \wedge \mathtt{X} > 0\}$
    $\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} > 0\}$
    `T = S;`
    $\{\mathtt{T} = h(n - \mathtt{X}) \wedge \mathtt{X} > 0\}$
    $\{1 = 1 \wedge \mathtt{T} = h(n - \mathtt{X}) \wedge \mathtt{X} > 0\}$
    `S = 1;`
    $\{\mathtt{S} = 1 \wedge \mathtt{T} = h(n - \mathtt{X}) \wedge \mathtt{X} > 0\}$
    $\{\mathtt{S} = 2^{h(n-\mathtt{X})-\mathtt{T}} \wedge \mathtt{X} > 0\}$
    `while (T!=0) (`
        $\{\mathtt{S} = 2^{h(n-\mathtt{X})-\mathtt{T}} \wedge \mathtt{X} > 0 \wedge \mathtt{T} \neq 0\}$
        $\{2 * S = 2^{h(n-\mathtt{X})-(\mathtt{T}-1)} \wedge \mathtt{X} > 0\}$
        `S = 2 * S;`
        $\{\mathtt{S} = 2^{h(n-\mathtt{X})-(\mathtt{T}-1)} \wedge \mathtt{X} > 0\}$
        `T = T - 1;`
        $\{\mathtt{S} = 2^{h(n-\mathtt{X})-\mathtt{T}} \wedge \mathtt{X} > 0\}$
    `)`
    $\{\mathtt{S} = 2^{h(n-\mathtt{X})-\mathtt{T}} \wedge \mathtt{X} > 0 \wedge \neg \mathtt{T} \neq 0\}$
    $\{\mathtt{S} = 2^{h(n-\mathtt{X})} \wedge \mathtt{X} > 0\}$
    $\{\mathtt{S} = h(n - (\mathtt{X} - 1)) \wedge \mathtt{X} - 1 \geq 0\}$
    `X = X - 1;`
    $\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} \geq 0\}$
`)`
$\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} \geq 0 \wedge \neg \mathtt{X} > 0\}$
$\{\mathtt{S} = h(n - \mathtt{X}) \wedge \mathtt{X} = 0\}$
$\{\mathtt{S} = h(n)\}$

Zadanie 2. Szukaną gramatyką jest $G = \langle \Sigma, \{S, W, A, T, E\}, S, P \rangle$, gdzie

$$P = \{S \to WAT, W \to WD, W \to E, DA \to AAD, DT \to T, EA \to aE, ET \to \epsilon\}$$

„Działanie” gramatyki G jest następujace: z S wyprowadzamy słowo WAT. Z symbolu W wyprowadzamy słowo ED^n. Każdy symbol D przesuwa się na prawo i „przeskakując” przez symbole A podwaja ich ilość. Po dojściu do końca słowa, tj. napotkaniu symbolu T, symbol D „znika”. Przesuwając w ten sposób symbole D generujemy słowo $EA^{2^n}T$. Na koniec symbol E przesuwa się w prawo, zamieniając każdy symbol A na symbol terminalny a. Po dojściu do końca słowa, tj. napotkaniu symbolu T, symbol E znika wraz z nim i wyprowadzenie słowa a^{2^n} jest zakończone.

Aby generować słowa postaci a^{n^2} możemy nieco zmodyfikować powyższą gramatykę. Zauważmy, że $(i+1)^2 = i^2 + (2i+1)$. Rozważmy słowa w_i zbudowane z symboli A i a w których liczba symboli a wynosi i^2, symboli A zaś $2i+1$. Dla $i = 0$ jest tylko jedno takie słowo $w_i = A$. Aby z w_i otrzymać w_{i+1}, należy dodać $2i+1$ symboli a, tj. tyle, ile jest symboli A w słowie w_i oraz dwa symbole A. Podobnie jak w poprzedniej gramatyce może do tego służyć specjalny symbol D, który przesuwa się na prawo i „przeskakuje” przez symbole a i A, przy czym mijając każdy symbol A dodaje jeden symbol a. Po dotarciu do końca słowa dodaje dwa symbole A i „znika”. Na koniec na prawo przesuwa się symbol E usuwając wszystkie symbole nieterminalne ze słowa. Mamy więc wyprowadzenie $S \Rightarrow WAT \overset{*}{\Rightarrow} ED^nAT \overset{*}{\Rightarrow} Ew_nT \overset{*}{\Rightarrow} a^{n^2}$, a szukaną gramatyką jest $G = \langle \Sigma, \{S, W, A, T, E\}, S, P\rangle$, gdzie

$$\begin{aligned} P \;=\; & \{S \to WAT, W \to WD, W \to E, DA \to aAD, DT \to AAT, \\ & EA \to E, Ea \to aE, ET \to \epsilon\} \end{aligned}$$

Zadanie 3. Jeśli pewne słowo zawiera przynajmniej jeden symbol a i przynajmniej jeden symbol m, to albo zawiera podciąg ma, albo am, tj. jest postaci $umav$ lub $uamv$ i redukuje się do uv. Nie jest więc w postaci normalnej. Z drugiej strony każde słowo złożone wyłacznie z symboli m lub a nie zawiera ani podciągu am, ani ma, jest więc w postaci normalnej. Niech

$$\overline{n} \;=\; \begin{cases} m^n & \text{gdy } n > 0 \\ \epsilon & \text{gdy } n = 0 \\ a^{-n} & \text{gdy } n < 0 \end{cases}$$

Zbiorem słów w postaci normalnej jest więc $\mathrm{NF} = \{\overline{n} \mid n \in \mathbb{Z}\}$.
Lemat 1: jeżeli $u \to v$, to $|u| = |v| + 2$. Istotnie, anihilacja pary am i ma zmniejsza długość słowa o dwa.
Wniosek: Nie istnieje nieskończony ciąg redukcji $u_0 \to u_1 \to u_2 \to \ldots$ i każde słowo posiada postać normalną.

Niech

$$\begin{aligned} \text{bilans}(\epsilon) &= 0 \\ \text{bilans}(um) &= \text{bilans}(u) + 1 \\ \text{bilans}(ua) &= \text{bilans}(u) - 1 \end{aligned}$$

Łatwo pokazać, że dla dowolnych słów u i v zachodzi $\text{bilans}(uv) = \text{bilans}(u) + \text{bilans}(v)$ (dowód przez indukcję względem u).
Lemat 2: jeżeli $u \overset{*}{\to} v$, to $\text{bilans}(u) = \text{bilans}(v)$. Istotnie, jeśli $u \overset{*}{\to} v$, to istnieją słowa $u_0 = u, u_1, \ldots, u_n = v$, takie, że $u_i \to u_{i+1}$, dla $i = 0, \ldots, n-1$. Wystarczy więc pokazać, że jeśli $u \to v$, to $\text{bilans}(u) = \text{bilans}(v)$, a teza wyniknie przez indukcję względem n. Jeśli $u \to v$, to istnieją słowa x i y, takie, że $v = xy$ i $u = xamy$ lub $u = xmay$. W pierwszym przypadku ponieważ $\text{bilans}(am) = 0$, to $\text{bilans}(v) = \text{bilans}(xy) = \text{bilans}(x) + \text{bilans}(y) = \text{bilans}(x) + \text{bilans}(am) + \text{bilans}(y) = \text{bilans}(xamy) = \text{bilans}(u)$. Podobnie w drugim, skoro także $\text{bilans}(ma) = 0$. To kończy dowód lematu.
Wniosek: Każde słowo posiada dokładnie jedną postać normalną. Istotnie, jeśli v_1 i v_2 są postaciami normalnymi słowa u, to na mocy lematu 2, skoro $u \overset{*}{\to} v_1$ i $u \overset{*}{\to} v_2$, to $\text{bilans}(u) = \text{bilans}(v_1) = \text{bilans}(v_2)$. Ale skoro $v_1, v_2 \in \mathrm{NF}$, to v_1 i v_2 są ciągami wyłacznie symboli m lub a, mają więc równy bilans tylko wtedy, gdy są równe, więc $v_1 = v_2 = \overline{\text{bilans}(u)}$. Zatem jedyną postacią normalną słowa u jest słowo $\overline{\text{bilans}(u)}$.
Własność Church-Rossera jest również konsekwencją lematu 2: jeśli $u \overset{*}{\to} v$ i $u \overset{*}{\to} w$, to $\text{bilans}(u) = \text{bilans}(v) = \text{bilans}(w)$. Niech $z = \overline{\text{bilans}(u)}$. Oczywiście $v \overset{*}{\to} z$ i $w \overset{*}{\to} z$.

Z własności Churcha-Rossera wynika, że $u \sim v$ wtedy i tylko wtedy, gdy istnieje słowo w, takie, że $u \xrightarrow{*} w$ i $v \xrightarrow{*} w$. Istotnie, $u \sim v$, jeśli istnieje ciąg słów $u_0 = v, u_1, \ldots, u_n = v$, takich, że $u_i \to u_{i+1}$ lub $u_{i+1} \to u_i$ dla $i = 0, \ldots, n-1$. Jest to równoważne stwierdzeniu, że istnieją ciągi $u_0 = u, \ldots, u_n = v$ oraz $v_1, \ldots, v_n$, takie, że $u_i \xrightarrow{*} v_{i+1}$ oraz $u_{i+1} \xrightarrow{*} v_{i+1}$, dla $i = 0, \ldots, n-1$. Przez indukcję względem n pokazujemy, że oznacza to, iż istnieje słowo w, takie, że $u \xrightarrow{*} w$ i $v \xrightarrow{*} w$.
Wniosek: $u \sim v$ wtedy i tylko wtedy, gdy $\text{bilans}(u) = \text{bilans}(v)$. Istotnie, $u \sim v$ wtedy i tylko wtedy, gdy istnieje słowo w, takie, że $u \xrightarrow{*} w$ i $v \xrightarrow{*} w$. Jeżeli takie słowo istnieje, to $\text{bilans}(u) = \text{bilans}(v)$. Z drugiej strony, jeśli $\text{bilans}(u) = \text{bilans}(v)$, to u i v mają tę samą postać normalną $w = \overline{\text{bilans}(u)} = \overline{\text{bilans}(v)}$, zatem odpowiednie słowo w istnieje.

Niech $D = \mathbb{Z}$ i $[\![u]\!] = \text{bilans}(u)$, dla każdego $u \in \{m, a\}^*$. Na mocy ostatniego wniosku $[\![u]\!] = [\![v]\!]$ wtedy i tylko wtedy, gdy $u \sim v$.

Zadanie 4. Jeżeli dla pewnego podstawienia θ jest $t\theta = s\theta$, to dla każdego $\theta' \geq \theta$ jest $t\theta' = s\theta'$. Istotnie, skoro $\theta' \geq \theta$, to istnieje podstawienie ρ, takie, że $\theta' = \theta\rho$. Zatem $t\theta' = t(\theta\rho) = (t\theta)\rho = (s\theta)\rho = c(\theta\rho) = s\theta'$.

Rozważmy zadanie

$$\{s_i = s_i'\}_{i=1}^n \cup \{t_j \neq t_j'\}_{j=1}^m. \tag{1}$$

Jeżeli klasyczne zadanie unifikacji $\{s_i = s_i'\}_{i=1}^n$ nie ma rozwiązania, to całe zadanie również nie może mieć rozwiązania. W przeciwnym razie niech θ_0 będzie najogólniejszym unifikatorem układu $\{s_i = s_i'\}_{i=1}^n$. Każde rozwiązanie zadania (1) musi być unifikatorem $\{s_i = s_i'\}_{i=1}^n$, jeśli więc θ jest rozwiązaniem (1), to $\theta \geq \theta_0$. Jeśli $t_j\theta_0 \neq t_j'\theta_0$ dla $j = 1, \ldots, m$, to θ_0 jest rozwiązaniem (1). W przeciwnym razie (1) nie posiada rozwiązania, bo skoro istnieje j, takie, że $t_j\theta_0 = t_j'\theta_0$, to dla każdego $\theta \geq \theta_0$ zachodzi także $t_j\theta = t_j'\theta$ i θ nie jest rozwiązaniem (1).

Aby więc znaleźć rozwiązanie (1) można wykonać następujacy algorytm:

1. Znajdź najogólniejszy unifikator θ_0 układu $\{s_i = s_i'\}_{i=1}^n$. Jeżeli takowy nie istnieje, to zadanie (1) również nie posiada rozwiązania.
2. Sprawdź, czy $t_j\theta_0 \neq t_j'\theta_0$ dla $j = 1, \ldots, m$. Jeżeli tak, zwróć θ_0. W przeciwnym razie zadanie (1) nie posiada rozwiązania.

Wyznaczone w ten sposób podstawienie jest najogólniejsze w tym sensie, że dla każdego innego rozwiazania θ zadania (1) zachodzi $\theta \geq \theta_0$. Niestety nie każde podstawienie $\theta \geq \theta_0$ jest rozwiazaniem (1). Dla przykładu rozwiązaniem $\{x \neq y\}$ jest podstawienie identycznościowe []. Ponadto $[x/c, y/c] \geq []$, jednak $[x/c, y/c]$ nie jest rozwiązaniem $\{x \neq y\}$.