Teoretická informatika - Úkol č.2

Buchal Petr, xbucha02

1. Prosinec, 2018

1. Příklad

1.1 Zadání

Mějme gramatiku $G = (\{S\},\ \{[,]\},\ P,\ S)$ s pravidly

$$S \to \epsilon |\ [S]S.$$

Nechť L je Dyckův jazyk nad jednou dvojicí závorek $\{[,]\}$. Dokažte, že $L \subseteq L(G)$. Postupujte takto:

(a) Ukažte, že každé slovo $w \in L$ lze napsat ve formě $[u]v$ kde $u, v \in L$. (Nápověda: Jak se dá napsat nejkratší neprázdný prefix slova w, který je sám také v L?)

(b) Dokažte, že $L \subseteq L(G)$, a to indukcí k počtu [ve slově. Tvrzení (a) použijte v indukčním kroku. (Báze: důkaz pro 0; Indukční krok: důkaz pro $i > 0$ za předpokladu, že tvrzení platí pro všechna $j < i$.)

1.2 Řešení

1.2.1 Bod a

1.2.2 Bod b

2. Příklad

2.1 Zadání

Je jazyk $L_{primes} = \{a^n | n \text{ je prvočíslo}\}$ bezkontextový? Dokažte.

2.2 Řešení

- Předpokládejme, že $L_{primes} \in \mathcal{L}_2$.
- $\exists k > 0 : \forall z \in L : |z| \geq k \Rightarrow \exists u, v, w, x, y \in \Sigma^* : z = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L$
- Uvažme libovolné $k > 0$ splňující uvedené tvrzení.
- Zvolme $z = a^b$, kde b je prvočíslo, které je rovno k nebo je nejbližším prvočíslem větším než k, $|z^b| = b \geq k$.
- Tedy $\exists u, v, w, x, y \in \Sigma^* : z = uvwxy \wedge vx \neq \epsilon \wedge |vwx| \leq k \wedge \forall i \geq 0 : uv^iwx^iy \in L$.
- Pokud v a x vyjádříme tak, že $v = a^c$ a $x = a^d$ kde $c \neq 0 \vee d \neq 0$, tak bude splněna podmínka $vx \neq \epsilon$, protože $c + d > 0$.
- Délku $|uwy|$ si můžeme vyjádřit jako $e = |uwy| - |v| - |x|$.
- Když zvolíme i rovno e, dostaneme slovo délky $|uv^ewx^ey| = e + c * e + d * e$
- Z tohoto tvaru vytkneme na $e(1 + c + d)$, pokud $e \neq \{0, 1\}$ délka slova není prvočíslo, protože je dělitelná e a zároveň $(1 + c + d)$, tedy slovo nepatří do jazyka.
- Pro $e = 0$ zvolíme $i = 2$, délka výsledného slova bude sudá a tudíž dělitelná 2 a slovo nebude náležet do jazyka.
- V případě $e = 1$, bude k délce $|vx|$ vždy přičtená 1, díky které se budou objevovat případy, kdy délka slova nebude dělitelná dvěma. Rovněž nelze využít první vztah, protože by se délka slova nezměnila a zůstala by prvočíslem. Zkusme provést úpravu, aby slovo bylo mocninou.

- Úpravu zkusíme provést pro mocninu dvou

$$b^2 = 1 + (c * i) + (d * i)$$

$$b^2 = 1 + i * (c + d), c + d = b - 1$$

$$b^2 = 1 + i * (b - 1)$$

$$\frac{b^2 - 1}{b - 1} = i$$

$$\frac{(b - 1)(b + 1)}{b - 1} = i$$

$$i = b + 1$$

- Pro $i = (b + 1)$, kde b je délka původního slova, nebude nové slovo patřit do jazyka, neboť je jeho délka druhá mocnina původní délky, což není prvočíslo.
- Ve všech rozděleních lze nalézt takové i, že $|uv^i wx^i y| \notin L_{primes}$. Nastává tedy spor a $L \notin \mathcal{L}_2$.

3. Příklad

3.1 Zadání

Nechť $a_0, a_1 \in N \backslash \{0\}$ jsou dané konstanty. Uvažujte jazyk

$$Affine = \{w \in \{0,\ 1\}^*\ |a_0 \cdot \#_0(w) + a_1 \cdot \#_1(w) - a_0 a_1 = 0\}.$$

Dokažte pomocí redukce z problému členství (membership problem), že problém, zda jazyk daného Turingova stroje M obsahuje alespoň jeden řetězec z jazyka *Affine*, je nerozhodnutelný. Dále uveďte ideu důkazu, že problém je částečně rozhodnutelný.

3.2 Řešení

4. Příklad

4.1 Zadání

Uvažujte programovací jazyk RationalC s následující gramatikou:

$$stmt = \mathrm{x} = \lceil \mathrm{odd(x)} \rceil | \mathrm{x} = \lfloor even(\mathrm{x}) \rfloor | \mathrm{x}\star = 2|\mathrm{x}/ = 2|return\ b|$$

if x % 2 == b goto n $\{$*stmt-list* $\rangle$ ::= $\{$*stmt* ; $\{$*stmt-list* $\rangle|\{$*stmt*$\rangle$

$\langle$*program*$\}$ ::= $\{$*stmt-list* ; return b_{i}

kde $n \in N, b \in \{0,\ 1\}$ a počátečn$ıneterminálje(program\}(uvažujemeže0\in\ N)$. Sémantikaje následuj$ıcı$:• Program v RationalC je spouštěn na stroji sjedn$ımregistrem$x, $jenžmužeobsahovatracionálnıčıslosneome$

• Na začátku běhu programu je v registru x uloženo přirozené $čıslo$x$_0 \in N$ (vstup programu).

• P$řıkaz$x = $\lceil$odd(x)$\rceil$ změn$ıceloučástčıslavregistruxnanejbližšıvětšıneborovnélichéčıslo.Např.$$\lceil$odd (42.1337) $\rceil$ = 43.1337a$\lceil$odd (1.00777) $\rceil$ = 1.00777.

• P$řıkaz$x = $\lfloor$even (x)$\rfloor$ změn$ıceloučástčıslavregistruxnanejbližšımenšıneborovnésudéčıslo.Např.$$\lfloor$even (42.1337) $\rfloor$ = 42.1337a$\lfloor$even (1.00777) $\rfloor$ = 0.00777.

• P$řıkaz$x $\star = 2$ vynásob$ıčıslovregistruxdvěma.$

• P$řıkaz$x /= 2$vydělıčıslovregistruxdvěma.$

• P$řıkazreturnbukončıprogramsnávratovouhodnotou$b.

• P$řıkazif$x%2 == b$gotonprovedepodmıněnýskoknan-týpřıkaz(přıkazyjsoučıslványod1aoddělenyznakemstředn$

$= 1)$dělitelnádvěma.Uvažujte, žesyntaktickysprávnýprogramneobsahujeskoky, kdenjevětšıneždčısloposlednıhopřıkaz$

Obrázek 1 obsahuje p$řıkladyprogramuvjazyceRationalC.$

Dokažte, že programovac$ıjazykRationalCjeTuringovskyúplný, tj., dokažte, že$

P$řıklad1 : programvracejıcı1právětehdy, kdyžje$x$_0$ dělitelné 8

P$řıklad1 : programvracejıcı1právětehdy, kdyžje$x$_0$ dělitelné 8

P$řıklad2 : programvracejıcı1právětehdy, kdyžbinárnızápisčısla$x$_0$ patř$ıdojazykapopsaného$ regulárn$ımvýrazem$(0+1)*0011$(01 + 10)^*$.

0 if x % 2 == 1 goto *6*;
1 x/ = 2_j0 if x % 2 == 0 goto *3*;
2 if x % 2 == 1 goto *6*; 1 if x % 2 == 1 goto *3*;
3 x/ = 2_j 2 x/ = 2_j // *loop* head 4 if x % 2 == 1 goto *6*; 3 if x % 2 == 1 goto *7*;
5 return 1; 4 x/ = 2_j
6 return 0_j 5 if x % 2 == 1 goto *2*;
6 if x % 2 == 0 goto 14;

7 x / = 2_j

8 if x % 2 == 0 goto *2*;
9 x/ = 2_j // found ′11′ 10 if x % 2 == 1 goto 14;

11 x / = 2_j

12 if x % 2 == 1 goto 14;
13 return 1;
14 return 0_j
Obrázek 1: P ř*ıkladyprogramuvjazyceRationalC*

(a) pro každý TS M nad abecedou $\{0,\ 1\}$ a řetězec $w \in \{0,\ 1\}^*$ lze sestrojit program P_Mv jazyce RationalC a zvolit počátečn*ıhodnotu*x_0 tak, že P_M skonč*ısnávratovouhodnotou*1*právětehdy, když*w∈ $L(M)$;

(b) pro každý program P vjazyce RationalC a počátečn*ıhodnotu*x_0 lze sestrojit TS M_P a řetězec $w \in \{0,\ 1\}^*$tak, že $w \in L(M_P)$ právě tehdy, když Ps počátečn*ıhodnotou*x_0 skonč*ısnávratovouhodnotou*1.

Nápověda: binárn*ızápisčısla*42.625_{10} je 101010.1012.

4.2 Řešení