1) Zdeněk Riegel

δ:

$(Q_0, a) \to (\vec{Q}, a, 0)$
$(Q_0, b) \to (\vec{Q}, b, 0)$
$(Q_0, \square) \to (\vec{Q}, \square, 0)$
} ~~[illegible]~~ toto by tu nemuselo být pokud by $\vec{Q} = Q_0$, je to jen pro přehlednost

$(\vec{Q}, a) \to (\vec{Q}, a, +1)$
$(\vec{Q}, b) \to (\vec{Q}, b, +1)$
} posun přes všechny znaky do prava

4

$(\vec{Q}, \square) \to (\overleftarrow{Q}, \triangleleft, -1)$ - za slovem (vpravo) dopíši $\triangleleft$ a začnu procházet doleva

$(\overleftarrow{Q}, a) \to (\overleftarrow{Q}, a, -1)$
$(\overleftarrow{Q}, b) \to (\overleftarrow{Q}, b, -1)$
} posun přes všechny znaky do leva

$(\overleftarrow{Q}, \square) \to (Q_F, \triangleright, +1)$ - před slovem (vlevo) dopíši $\triangleright$, posunu se o 1 doprava aby byla hlava na začátku slova a skončím.

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, \square, \triangleleft, \triangleright\}$, $Q = \{Q_0, \overleftarrow{Q}, \vec{Q}, Q_F\}$

ukázka výpočtu: $Q_0 baa \vdash \vec{Q} baa \vdash b\vec{Q}aa \vdash ba\vec{Q}a \vdash baa\vec{Q} \vdash ba\overleftarrow{Q}a\triangleleft \vdash$
$\vdash b\overleftarrow{Q}aa\triangleleft \vdash \overleftarrow{Q}baa\triangleleft \vdash \overleftarrow{Q}\square baa\triangleleft \vdash \triangleright Q_F baa\triangleleft$

2) jednotlivé znaky z Γ budu kódovat n-ticemi 0 a 1, kde n záleží na počtu znaků Γ, při běhu tohoto stroje, budu při čtení n-tic ukládat informaci o přečtených znacích n-tice do stavu

~~pokud n=3 a A → 010 (kódování)~~

2,5

~~pak instrukce $(Q_0, A) \to (Q_0, A, +1)$~~

↓

~~Q_0^0 010 +~~

4) ~~a) Pro každou instanci problému NHP platí že algoritmus, který řeší tento problém se zastaví na všech kladný~~

Existuje

a) Algoritmus který řeší tento problém se zastaví na všech kladných instancích (tedy když bude výpočet ~~na~~ M na w nekonečný) a odpoví 'Ano', v opačném případě se nezastaví

2

b) A ... algoritmus pro P

A_{co} ... algoritmus pro co-P

a

A se zastaví v kladných případech P

A_{co} se zastaví v kladných případech co-P, tedy v záporných případech P

Můžeme tedy vytvořit algoritmus B, který pro stejný vstup souběžně spustí algoritmy A i A_{co}

v případě že se zastaví A, zastaví se i B a vrátí ~~stejn~~ výstup A

v případě že se zastaví ~~B~~ A_{co}, zastaví se i B a vrátí opačný výstup

B se musí nutně zastavit na všech vstupech, protože v kladných případech P se zastaví A a v záporných A_{co}

B se tedy zastaví v každé instanci P a správně odpoví, tedy rozhoduje P a P je tím pádem rozhodnutelné

c) problém HP je částečně ~~ne~~ rozhodnutelný a NHP je co-HP, tedy pokud by co-HP byl částečně rozhodnutelný musel by být HP rozhodnutelný a to je spor.

4,5

5) nerozhodnutelnost z Riceovy věty neplyne v:

3

c) není triviální - např TS který z q_0 vždy ihned přejde do q_F nikdy neudělá více jak 1 krok

není I/O vlastnost - např pro 2 TS identity (mají stejný výstup jako vstup) určitě mají stejnou I/O tabulku

ale 1. může z $Q_0 \to Q_F$ a tedy vlastnost nemá

a 2. může přejít z $Q_0 \to Q_{1000}$ a v každém kroku přecházet z $Q_i \to Q_{i-1}$ ana $Q_1 \to Q_F$, tedy provede > 100 kroků a vlastnost má

d) vlastnost je triviální, z definice TS je každá pásková abeceda konečná

a) b) e) pak nerozhodnutelnost z riceovy věty plyne

b) pokud jsou tabulky 2 TS stejné, pak v obou je 10×0 nebo ne, tedy vlastnost mají oba nebo ani jeden, je to I/O vlastnost a není triviální

e) podobně jako b

3)

~~L - kódy gram, které generují $\{1,0\}^*$, ale ne svůj kód~~

~~G_x jiz C a negen. vlastní kód~~

předp: gramatika $G \in C$ a $L(G) = L$

~~$G \in C$ a gener $L(G) = L$~~

důkaz: pokud $\langle G \rangle \in L$, tak $\langle G \rangle \in L(G)$ a nemůže být v L - spor

pokud $\langle G \rangle \notin L$, tak $\langle G \rangle \notin L(G)$ a $G \in C$, tedy G musí být v L - spor

3,5

Zdeněk Riegel

17,5

První zápočtová písemka (verze 21b) (max. 20 bodů)

1. (4 body) Navrhněte přehledně (jednopáskový) Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, \{q_F\})$ podle následující specifikace: $Q = \{q_0, q_F, \dots\}$ ... doplňte další stavy, které použijete. $\Sigma = \{a, b\}$. $\Gamma = \{a, b, \Box, \triangleright, \triangleleft, \dots\}$ ... doplňte další páskové symboly, které použijete. Pro počáteční konfiguraci q_0w (kde $w \in \Sigma^*$) výpočet skončí v konfiguraci $\triangleright q_F w \triangleleft$. Funkci δ zapište **přehledně** jako sadu instrukcí; místo $\delta(q, x) = (q', y, d)$ pište $(q, x) \to (q', y, d)$ (zde $d \in \{-1, 0, +1\}$ určuje posun hlavy). Nutné jsou **vysvětlující komentáře** umožňující snadno pochopit, jaký algoritmus vlastně realizujete. Nakonec ukažte výpočet vašeho stroje M na slově baa, jako posloupnost konfigurací $q_0baa \vdash \cdots \vdash \cdots\cdots \vdash \triangleright q_F baa \triangleleft$.

2. (4 body) Vysvětlete, jak lze Turingův stroj $M = (Q, \{0, 1\}, \Gamma, \delta, q_0, \{q_F\})$ simulovat strojem $M' = (Q', \{0, 1\}, \{0, 1, \Box\}, \delta', q'_0, \{q'_F\})$ (omezili jsme tedy množinu páskových symbolů). Popište ideu konstrukce M' a podle časových možností pak popište M' co nejpřesněji.

3. (4 body) Uvažujme nějakou třídu $\mathcal{C}$ formálních gramatik s množinou terminálních symbolů $\{0, 1\}$ (gramatiky z dané třídy jsou tedy typu $G = (\Pi, \{0, 1\}, S, P)$); gramatika G generuje jazyk $L(G) \subseteq \{0, 1\}^*$. Uvažujme nějaké kódování gramatik pomocí řetězců z $\{0, 1\}^*$; k $G \in \mathcal{C}$ je přiřazen kód $\langle G \rangle \in \{0, 1\}^*$. Dokažte, že jazyk

$$L = \{\langle G \rangle \mid G \in \mathcal{C} \text{ a } \langle G \rangle \notin L(G)\}$$

není generován žádnou gramatikou z třídy $\mathcal{C}$.

4. (5 bodů) Uvažujte problém nezastavení, označený NHP:
Instance: Turingův stroj M a vstup w; *Otázka*: Je výpočet stroje M na w nekonečný?
a) Vysvětlete, co znamená tvrzení "Problém NHP je částečně rozhodnutelný".
b) Dokažte, že když je nějaký (ANO/NE) problém $\mathcal{P}$ částečně rozhodnutelný a zároveň je i jeho doplňkový problém (co-$\mathcal{P}$) částečně rozhodnutelný, tak $\mathcal{P}$ je rozhodnutelný.
c) Použijte tvrzení dokázané v b) k důkazu toho, že NHP není částečně rozhodnutelný.

5. (3 body) Uvažujme následující vlastnosti Turingových strojů (vlastnost je dána otázkou, na niž je odpověď buď ANO, tedy zadaný stroj M vlastnost má, nebo NE, tedy M vlastnost nemá):
1/0 a) Rozhoduje M problém ekvivalence konečných automatů ?
1/0 b) Vydá M alespoň pro deset různých vstupů výstup 0 ?
c) Existuje vstup, pro který M provede více než sto kroků ?
triv d) Je pásková abeceda stroje M konečná množina ?
1/0 e) Vydá M pro vstup 001 výstup 001001 ?
Uveďte a zdůvodněte případy, u nichž *neplyne* nerozhodnutelnost uvedené vlastnosti z Riceovy věty. Uveďte i případy, u nichž si nejste jisti (jsou-li jaké).