Lektion 6 och 7

Svar/lösningsförslag

1. (a) Körning på abb#aaa.

#abb#aaa

abb#aaa

abbxaaa

abbxaaa#

abbxaaay

#abbxaaay

abbxaaay

#bbxaaay

bbxaaay

bbxaaay

bbxaay#

#bbxaay

bbxaay

#bxaay

bxaay

bxaay

bxay#

#bxay

bxay

#xay

xay

xay

xy#

#xy

xy

xy

ja

TM:en JA suddar tapen och skriver 'ja'.

Om TM:en körs på ab#aaa så stannar den med svaret "nej". (Jag gör inre körningen.)

(b) TM:en besvarar frågan om u och v har samma längd. Dvs. om den startas i $\underset{\triangle}{\#}u\#v$ så stannar den med svaret "ja" om $|u| = |v|$ och med svaret "nej" annars.

2. (a)

$\underset{\triangle}{\#}11111$

$\underset{\triangle}{1}1111$

$\underset{\triangle}{\#}1111$

$\underset{\triangle}{1}111$

$\underset{\triangle}{\#}111$

$111\underset{\triangle}{\#}$

$111\#\underset{\triangle}{\#}$

$111\#\underset{\triangle}{1}$

$111\underset{\triangle}{\#}1$

$\underset{\triangle}{\#}111\#1$

$\underset{\triangle}{1}11\#1$

$\underset{\triangle}{\#}11\#1$

$\underset{\triangle}{1}1\#1$

$\underset{\triangle}{\#}1\#1$

$1\underset{\triangle}{\#}1$

$1\#1\underset{\triangle}{\#}$

$1\#1\underset{\triangle}{1}$

$1\underset{\triangle}{\#}11$

$\underset{\triangle}{\#}1\#11$

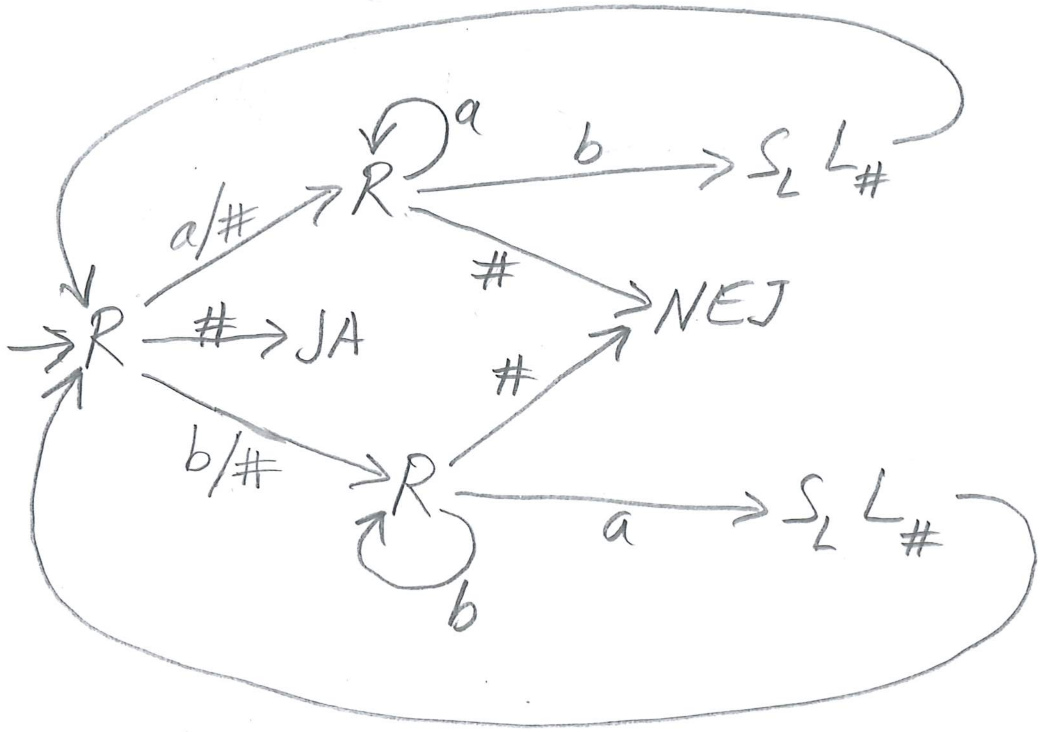
$\underset{\triangle}{1}\#11$

$\underset{\triangle}{\#}\#11$

$\underset{\triangle}{\#}11$ terminerar

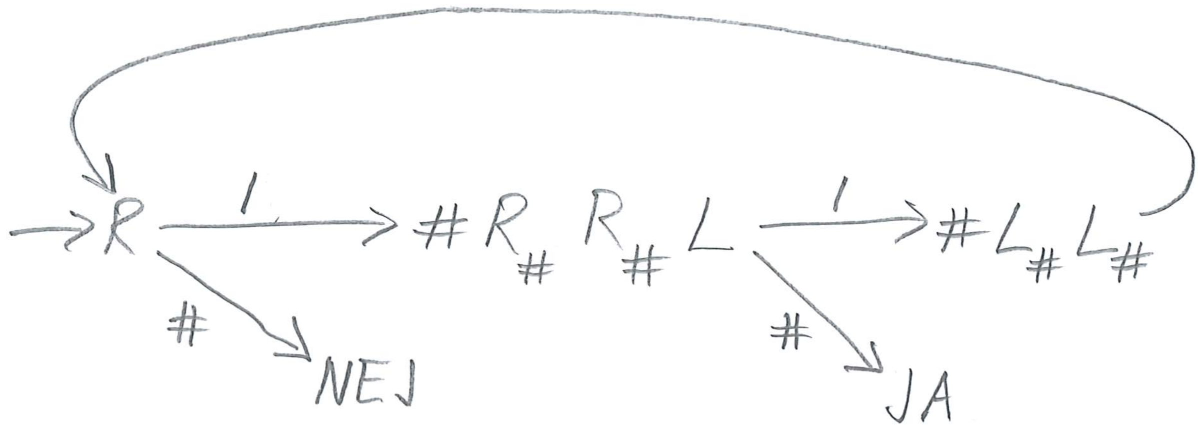
(b) TM:en beräknar heltalskvoten vid division med 2. Med andra ord, $f(n) = \lfloor \frac{n}{2} \rfloor$ där $\lfloor x \rfloor$ beräcknar det största heltalet som är mindre eller lika med x.

4. (a)



(b) Vi får en TM som accepterar samma språk om 'NEJ' ersätts med en oändlig loop, tex .

5.



7. reguljära $\subset$ sammanhangsfria $=$ PDA-accepterbara $\subset$ TM-avgörbara $\subset$ TM-accepterbara $=$ restriktionsfria.

9. (a) Låt

$\Omega = \{ L : L$ är TM-accepterbar och innehåller någon sträng av längd högst $8 \}$.

För att kunna använda Rices sats behöver vi <u>verifiera tre saker</u>:

- <u>Alla språk i Ω är TM-accepterbara</u>. Detta följer direkt från definitionen av Ω.

- Ω är inte tom, dvs innehåller minst ett språk. Detta stämmer för om (tex) $L = \{aa\}$ så $L \in \Omega$, för $\{aa\}$ är reguljärt och därmed TM-accepterbart.

- Ω innehåller inte alla TM-accepterbara språk. Detta stämmer för om (tex.) $L = \{a^9\}$ så $L \notin \Omega$. (och $\{a^9\}$ är reguljärt och därmed TM-accepterbart).

Nu följer från Rices sats att ingen TM kan avgöra för en godtycklig TM M om $L(M)$ innehåller någon sträng av längd högst 8.

(b) Låt

$$\Omega = \{L : L \text{ är TM-accepterbart och } \varepsilon \in L\}.$$

Verifiera nu att de tre punkterna i del (a) är uppfyllda, och hänvisa sedan till Rices sats.

(c) Låt

$$\Omega = \{L : L \text{ är TM-accepterbart och innehåller alla strängar (över } L\text{:s alfabet) av jämn längd}\}.$$

Verifiera att de tre punkterna i del (a) är uppfyllda och hänvisa sedan till Rices sats.

10.(a) Följande algoritm terminerar om och endast om en TM M (given som input) accepterar ε. (Från Church-Turings tes följer att algoritmen kan "översättas" till en TM.)

Kör den universella TM:en $\mathcal{U}$ på input M och ε (mer precist på input $K_M \# K_\varepsilon$ där K_M och K_ε är koderna för M och ε, respektive). Då kommer $\mathcal{U}$ att terminera om och endast om M terminerar på input ε. Med andra ord så terminerar $\mathcal{U}$ om och endast om M accepterar ε.

(b) Om $\overline{L}$ vore TM-avgörbar så vore även $\overline{\overline{L}} = L$ TM-avgörbar vilket vi har visat (i tidigare uppgift) att den <u>inte</u> är. Sålunda är $\overline{L}$ <u>inte</u> TM-avgörbar.

Från a-delen vet vi att L är TM-accepterbar. Om även $\overline{L}$ är TM-accepterbar så följer från en sats (Sats 7.2 i boken) att L är TM-avgörbar vilket vi har visat att den inte är. Alltså är $\overline{L}$ <u>inte</u> (ens) TM-accepterbar.

11. Enligt Church-Turings tes räcker det att beskriva informella algoritmer.

(a) Givet en godtycklig DFA M, kör M på alla strängar (över M:s alfabet) av längd ≤ 8. Det finns bara ändligt många sådana strängar och varje körning terminerar efter ändligt många steg. Därför får vi, efter ändlig många "steg" reda på om M accepterar någon sträng av längd ≤ 8.

(b) Givet en godtycklig DFA M, kör M på ε och se om ε accepteras.

(c) Kräver lite mer avancerade resonemang.

12. Svaret är ja.

Informell algoritm:

Givet en godtycklig TM M och sträng w, låt f(M, w) vara antalet konfigurationer där tapen innehåller (endast) w.
Kör M på w i f(M, w)+1 steg. Om M har skrivit över ett tecken med ett annat tecken så stanna och svara 'ja'. Annars så kommer M aldrig att skriva över ett tecken med ett annat tecken och vi stannar med svaret 'nej'. Detta pga att M är deterministisk och har befunnit sig i samma konfiguration två gånger, så om M inte har stannat efter f(M, w)+1 steg så kommer M bara att upprepa samma "slinga" utan att någonsin skriva över ett tecken med ett annat tecken.

13. Svaret är nej, men vi kan inte visa detta med Rices sats, för detta problem handlar om TM:ars beteende under körning (och inte endast om egenskaper hos TM:ars språk).

Antag, för att nå en motsägelse, att en TM N avgör problemet ifråga. Vi kan då konstruera en TM som avgör stopp-problemet (vilket motsäger att stopp-problemet är oavgörbart) så här (med hänvisning till Church-Turings tes):

<u>Algoritm</u>: Låt en godtycklig TM M och sträng w vara givna.
Modifiera M till M' där M' fungerar som M förutom att

- M' har en nytt tecken * som inte finns i M:s alfabet,
- när M skriver '#' så skriver M' '*' i stället,

- och '$*$' behandlas av M' precis som det vanliga blanktecknet $\#$ (så om M har en övergång $\delta(p,\#)=(q,\sigma)$ så har M' en övergång $\delta'(p,*)=(q,\sigma')$ där $\sigma'=\sigma$ om $\sigma\neq\#$ och $\sigma'=*$ om $\sigma=\#$),
- då M stannar så skriver M' i stället '$\#$' och stannar sedan. Dvs. '$\longrightarrow ⓞ$' ersätts med '$\longrightarrow \# \longrightarrow ⓞ$'.

Sedan körs N med M' och w som input. Nu är det så att

M stannar efter start på w $\iff$

M' stannar efter start på w $\iff$

M' skriver $\#$ någon gång (faktiskt exakt en gång) efter start på w.

Det följer att N svarar ja om M stannar efter start på w och N svarar nej annars. Vi låter vår algoritm svara ja om N svarar ja och nej om N svarar nej.
Då avgör vår algoritm stopp-problemet (vilket motsäger att det är oavgörbart).