Uitwerking tentamen Fundamentele Informatica 1 (I&E)
maandag 16 december 2013

07:55

1 (a) $L_1L_2 = \{abaa, abb, ababb, aa, b\}$

07:57

(b) Ja, het geldt altijd.

Laat $x \in L_2L_3$ willekeurig.

Dan is $x = x_2x_3$ met $x_2 \in L_2$ en $x_3 \in L_3$

Er geldt: $L_2^* = L_2^0 \cup L_2^1 \cup L_2^2 \cup L_2^3 \cup \ldots$

$\Longrightarrow L_2 \subseteq L_2^*$

Omdat $x_2 \in L_2$, is dan ook $x_2 \in L_2^* \Longrightarrow x_2 \in L_1 \cup L_2^*$.

Maar dan is $x_2x_3 \in (L_1 \cup L_2^*)L_3$.

Elk element $x \in L_2L_3$ zit dus ook in $(L_1 \cup L_2^*)L_3$

08:02

2 (a) $(aa^*b + bb^*a)^*$

08:03

(b) Als je alleen woorden x met $n_a(x)$ is even wil beschrijven,
kan dat met $b^*(ab^*ab^*)^*$

Nu moet je afdwingen dat er een keer bb voorkomt.
Het voorkomen van bb kan aan het begin zijn, of na de eerste a van een tweetal a's of na de tweede a van een tweetal a's.
Dat zijn dus drie mogelijkheden.
Bij de laatste twee mogelijkheden moeten we er rekening mee houden dat het betreffende tweetal a's nog voorafgegaan en gevolgd kan worden door andere tweetallen a's.
Dit levert de volgende reguliere expressie:

$$bbb^*(ab^*ab^*)^* + b^*(ab^*ab^*)^*(abbb^*ab^*)(ab^*ab^*)^*$$
$$+ b^*(ab^*ab^*)^*(ab^*abbb^*)(ab^*ab^*)^*$$

08:11

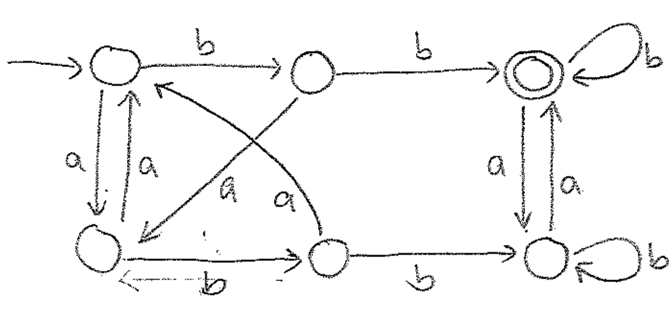
Deze expressie kan overigens flink ingekort worden, b.v. tot

$$\Big(bbb^* + b^*(ab^*ab^*)^*\big(abbb^*ab^* + ab^*abbb^*\big)\Big)(ab^*ab^*)^*$$

08:13

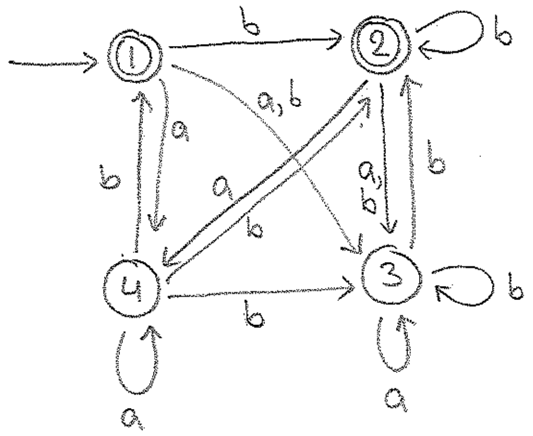
16:33

(c)



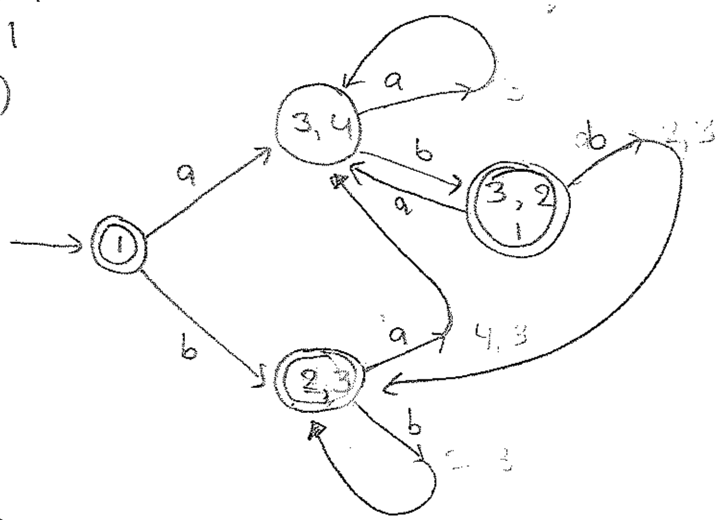
16:38

3(a)



15:4~~8~~4
21:21

(b)



21:30
21:31

4 (a)

De eerste vijf elementen van L_1 in de canonieke volgorde zijn:

aa, aaa, aaab, abaa (en vervolgens aaaaa, aaaab, abaaa)

aa: $j=1$; aaa: $j=2$; ↑ aaaa, $j=3$; aaab: $j=2, k=1$; abaa: $i=1$, $j=2$; aaaaa: $j=4$; aaaab: $j=3$, $k=1$; abaaa: $i=1$, $j=3$

21:36

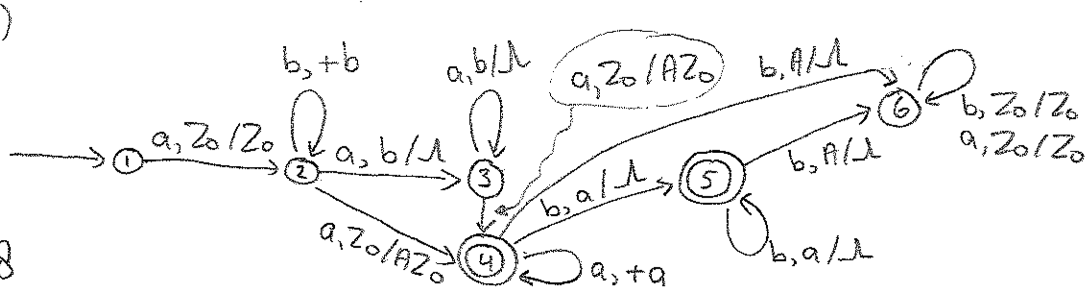
(b)

$S \rightarrow aBAC$	starten
$B \rightarrow bBa \mid \Lambda$	deelwoord van vorm $b^i a^i$ met $i \geq 0$
$C \rightarrow aCb \mid \Lambda$	deelwoord van vorm $a^k b^k$ met $k \geq 0$
$A \rightarrow aA \mid a$	deelwoord van vorm a^m met $m \geq 1$

Met $j = i+k+m$ weet je dan dat $j > i+k$.

21:40

(c)



Voor minder toestanden, zie blz 7

21:48

Uitwerking tentamen Fundamentele Informatica 1 (I&E)
maandag 16 december 2013

(d) De eerste zes elementen van L_1 in de canonieke volgorde

ab, abb, aabb, abab, abba, abbb (en vervolgens aabbb, ababb, abbab, abbba, abbbb)

$i=1$, $i=2$, $j=1, k=2$, $i=1, j=1, k=1$, $i=2, j=1$, $i=3$

21:52

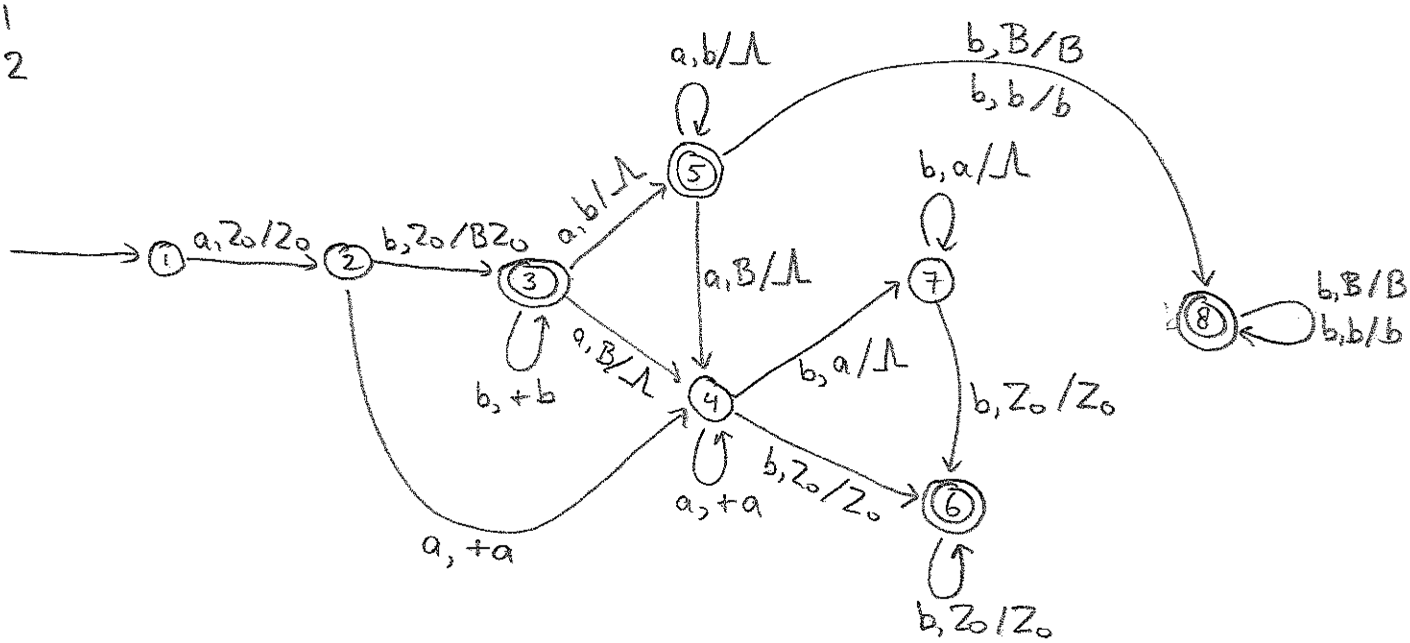
(e) $j < i+k \Longrightarrow j < i,\ k \geq 0$ of $j \geq i$, maar $i+k > j \Leftrightarrow k > j-i \Leftrightarrow k-(j-i) > 0$

(1) $\Leftrightarrow i-j > 0$ (2)

$S \rightarrow S_1 \mid S_2$	opstarten naar variant 1 of 2
$S_1 \rightarrow aBAC$	opzet variant 1
$B \rightarrow bB \mid b$	b^{i-j} met $i-j>0$
$A \rightarrow bAa \mid \Lambda$	$b^j a^j$ met $j \geq 0$
$C \rightarrow bC \mid \Lambda$	b^k met $k \geq 0$
$S_2 \rightarrow aDEF$	opzet variant 2
$D \rightarrow bDa \mid \Lambda$	$b^i a^i$ met $i \geq 0$
$E \rightarrow aEb \mid \Lambda$	$a^{j-i} b^{j-i}$ met $j-i \geq 0$
$F \rightarrow bF \mid b$	$b^{k-(j-i)}$ met $k-(j-i)>0$

22:01
22:02

(f)



22:10. Voor minder toestanden, zie blz 7

22:11 Uitwerking tentamen Fundamentele Informatica 1 (I&E) maandag 16 december 2013

5 (a)

De eerste zes elementen van L_1 in de canonieke volgorde zijn

a, aa, aaa, baa, $aaaa$, $abaa$ (en vervolgens $baaa$, $aaaaa$, $abaaa$, $baaaa$, $bbaaa$)

$k=1$, $k=2$, $k=3$, $j=1$ $k=2$, $k=4$, $i=1$, $j=1$ $k=2$

22:15 (b) Stel dat L_1 wel context-vrij is. Dan is er een getal n zodat elke $u \in L$ met $|u| \geq n$ weggepompt en opgepompt kan worden.

$u = a^n b^n a^n$ is niet geschikt, want zit niet in L

(want $j = n = k$, terwijl $j < k$ moet zijn)

$u = a^n b^n a^{2n}$ is niet geschikt, want er is een opsplitsing $vwxyz$ van u waarbij u zonder problemen weggepompt en opgepompt kan worden. Neem b.v. $v = a^n b^n$, $w = \Lambda$, $x = \Lambda$, $y = a$, $z = a^{2n-1}$

Dan geldt $vw^m xy^m z = a^n b^n a^m a^{2n-1}$, en dat woord zit voor elke $m \geq 0$ nog steeds in L_1 want $i = n$

$j = n$

$k = 2n - 1 + m > n$

($\Leftrightarrow n + m > 1$, maar daar mag je wel van uit gaan).

We nemen dus $u = a^n b^n a^{n+1}$. Deze u zit in L_1, en $|u| \geq n$.
Laat $vwxyz$ een willekeurige opsplitsing zijn van $u = a^n b^n a^{n+1}$ met $|wy| > 0$ en $|wxy| \leq n$.

want $|wxy| \leq n$

* als wy a's uit eerste serie bevat, bevat wy geen a's uit tweede serie. Bijpompen van wy ($m \geq 2$) zorgt er dan voor dat we vooraan minstens zoveel a's krijgen als achteraan $\Rightarrow$ niet in L.
* als wy geen a's uit eerste serie bevat, bevat wy

  (*) of b's uit midden

  (**) of a's uit tweede serie

  of beide.

Wegpompen van wy ($m = 0$) zorgt er dan voor dat we (geval (*)) minder b's in het midden krijgen dan a's vooraan (geval (**)) hoogstens zoveel a's achteraan overhouden als a's vooraan.

In beide gevallen zit het woord niet in L.

22:33 Conclusie: hoe de opsplitsing $vwxyz$ van $u = a^n b^n a^{n+1}$ er ook uitziet, we kunnen u óf niet oppompen óf niet wegpompen. Dat is strijdig met het pomplemma $\Rightarrow$ de veronderstelling dat L_1 context-vrij is, klopt niet.

22:34

6 (a)

V is de verzameling van variabelen (niet-terminalen)

Σ is de verzameling van terminalen

S is de startvariabele

Zowel V als Σ is een eindige verzameling

22:37

(b) Reguliere grammatica's:

er zijn twee mogelijke soorten producties:

$A \to \sigma B$ met $A, B \in V$ (eventueel hetzelfde) en $\sigma \in \Sigma$

$A \to \Lambda$ met $A \in V$

Context-vrije grammatica's

producties zijn van de vorm

$A \to \beta$ met $A \in V$ en $\beta \in (V \cup \Sigma)^*$

Context-vrije grammatica's in Chomsky normaalvorm

er zijn twee mogelijke soorten producties

$A \to BC$ met $A, B, C \in V$ (eventueel (deels) hetzelfde)

$A \to \sigma$ met $A \in V$ en $\sigma \in \Sigma$

Unrestricted grammatica's:

producties zijn van de vorm

$\alpha \to \beta$ met $\alpha, \beta \in (V \cup \Sigma)^*$, waarbij α minstens één variabele bevat

22:43

7 (a) De eerste drie elementen van L in de canonieke volgorde zijn

abcc, abccc, abbccc (en vervolgens abcccc, aabbccc, abbcccc, abccccc)

22:45

(b) T controleert eerst of x van de vorm $a^i b^j c^k$ is met $1 \le i, j, k$.

Als dat zo is, gaat T achtereen volgende a's markeren, en bij iedere a ook een b en een c (steeds de meest linkse a, b, c die nog niet gemarkeerd is). Markeren betekent: hoofdletter van maken.

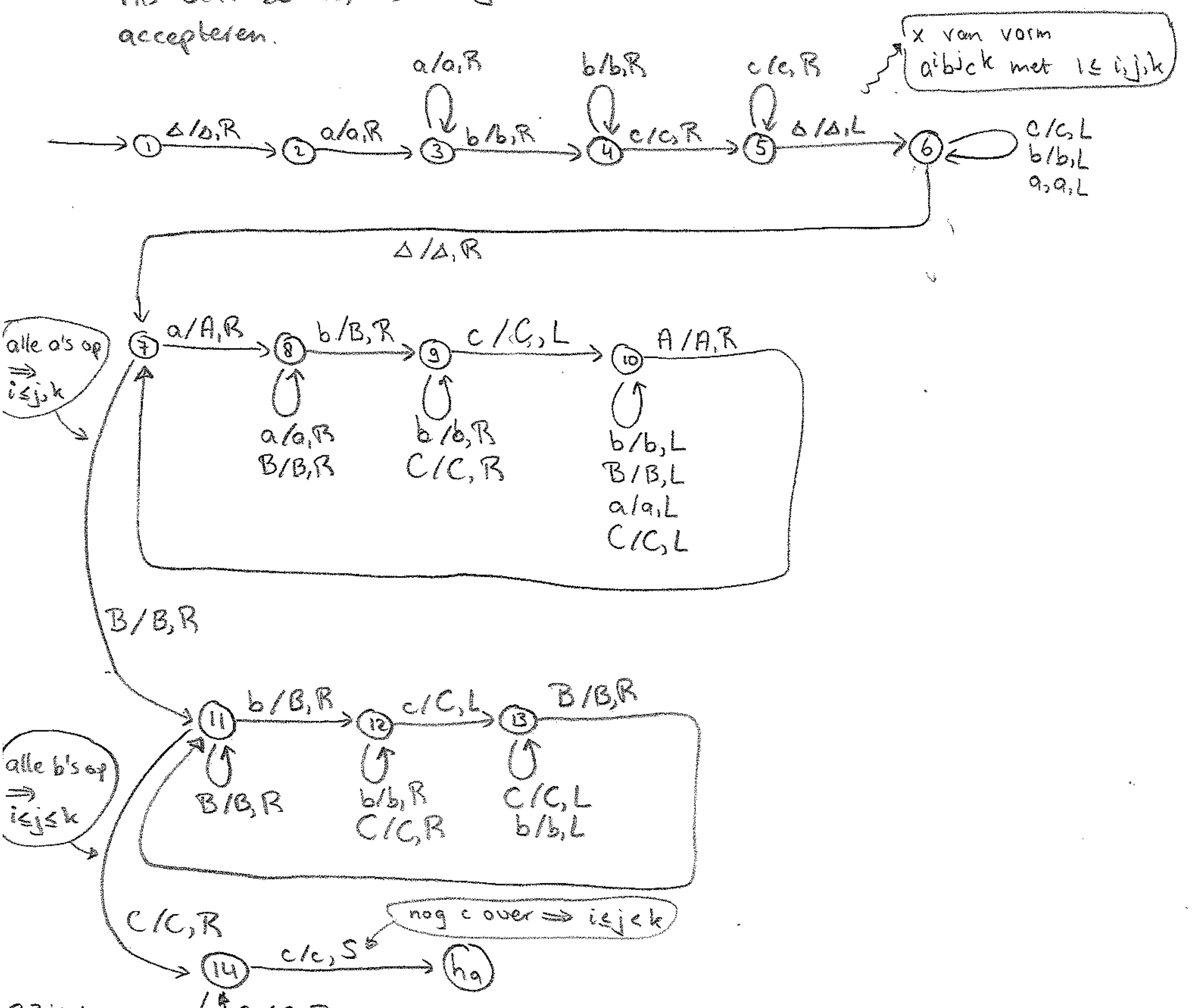
Als dat lukt bij elke a, is $i \le j, k$.

Vervolgens gaat T resterende b's markeren en bij elke b ook een c.

Als dat lukt bij elke b, is $i \le j \le k$.

Ten slotte controleert T of er nog minstens één ongemarkeerde c is.

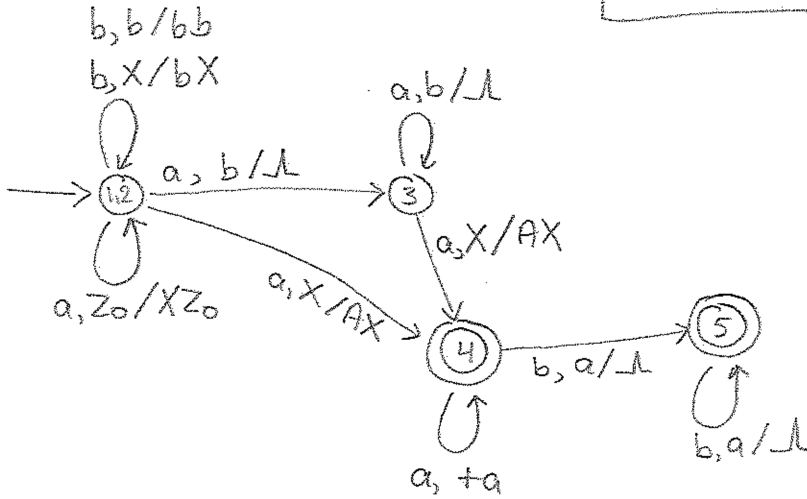
Als dat zo is, is $i \le j < k$, en zit x in L, zodat we mogen accepteren.



23:01

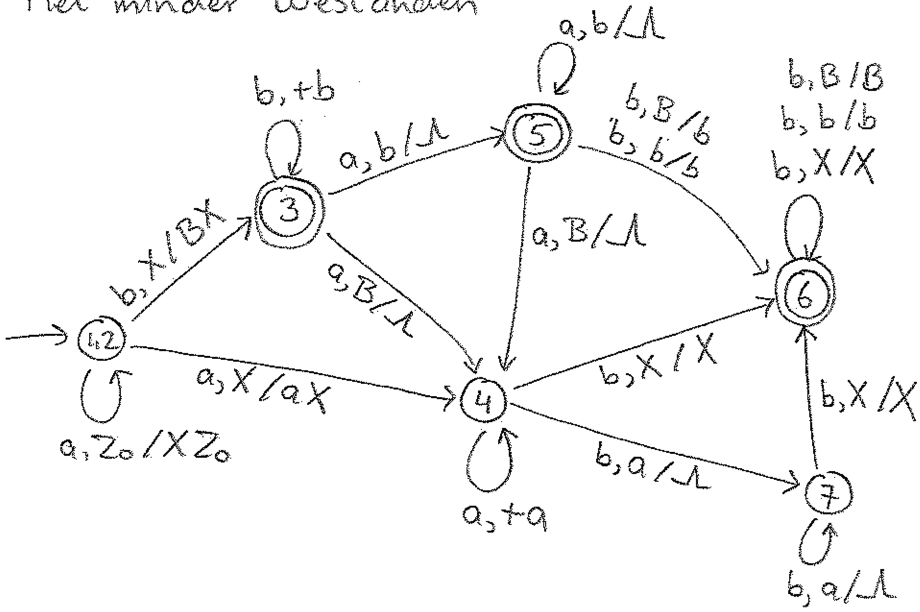
Uitwerking tentamen
Fundamentele Informatica 1 (I&E)
maandag 16 december 2013

2(c) Met minder toestanden:



Ten opzichte van blz. 2
* toestanden 1 en 2 samengevoegd m.b.v. speciaal symbool X
* toestand 6 geschrapt: we crashen gewoon.

2(f) Met minder toestanden



Ten opzichte van blz 3
* toestanden 1 en 2 samengevoegd m.b.v. speciaal symbool X
* toestanden 6 en 8 samengevoegd.