Prof. Dr. Christoph Karg
Studiengang Informatik
Hochschule Aalen

Klausur Berechenbarkeit & Komplexität (Sommersemester 2011)

Lösungshinweise

(Alle Angaben ohne Gewähr[1])

Aufgabe 1. Auswertung der Formel

$$F = (\neg x_1 \lor x_2) \land (x_3 \lor \neg x_2) \land (\neg x_1 \lor \neg x_3) \land (x_1 \lor x_3) \land (\neg x_1 \lor \neg x_2)$$

mit dem 2-SAT Algorithmus

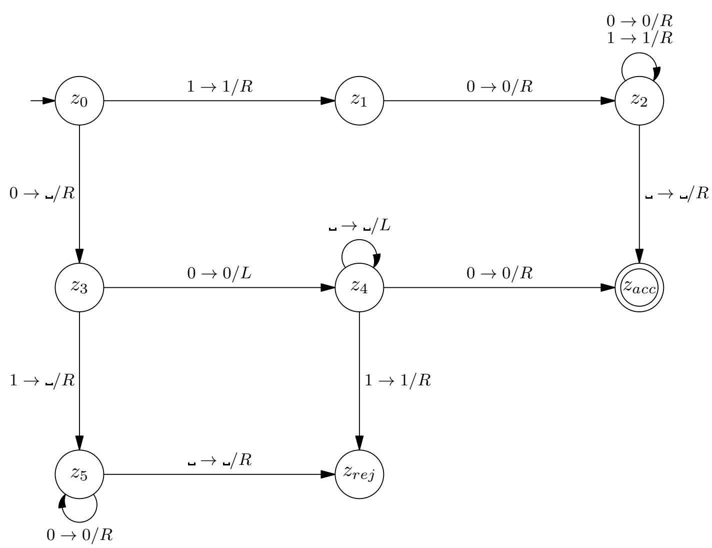
first	x_1	x_2	x_3	$\neg x_1 \lor x_2$	$x_3 \lor \neg x_2$	$\neg x_1 \lor \neg x_3$	$x_1 \lor x_3$	$\neg x_1 \lor \neg x_2$
true	1			$\rightsquigarrow x_2 = 1$				
true	1	1		$\checkmark$	$\rightsquigarrow x_3 = 1$			
true	1	1	1	$\checkmark$	$\checkmark$	$\lightning$		
false	0			$\checkmark$				
false	0			$\checkmark$		$\checkmark$		
false	0			$\checkmark$		$\checkmark$	$\rightsquigarrow x_3 = 1$	
false	0		1	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	
false	0		1	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
true	(0)		(1)	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$
true	(0)	1	(1)	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$
true	(0)	(1)	(1)	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$	$(\checkmark)$

Eine erfüllende Belegung für F ist $x_1 = 0$, $x_2 = 1$, $x_3 = 1$.

[1] Sachdienliche Hinweise zur Fehlerbekämpfung senden Sie bitte an `christoph.karg@htw-aalen.de`

Aufgabe 2.

a) Das Zustandsdiagramm der Turing Maschine M ist:



b) Konfigurationsfolge für die Eingabe $x = 010100$:

$$\begin{aligned} & \sqcup z_0 010100 \\ \to_M \; & \sqcup\sqcup z_3 10100 \\ \to_M \; & \sqcup\sqcup\sqcup z_5 0100 \\ \to_M \; & \sqcup\sqcup\sqcup 0 z_5 100 \end{aligned}$$

Die Turing Maschine bricht während der Verarbeitung der Eingabe ab. Die Eingabe x wird deshalb verworfen. Also ist $x \notin L(M)$.

c) M stoppt nicht auf allen Eingaben. Ein Beispiel ist $x = 001$. Die Konfigurationsfolge von M auf Eingabe x ist:

$$\begin{aligned} & \sqcup z_0 001 \\ \to_M \; & \sqcup\sqcup z_3 01 \\ \to_M \; & \sqcup z_4 \sqcup 01 \\ \to_M \; & \sqcup z_4 \sqcup\sqcup 01 \\ \to_M \; & \sqcup z_4 \sqcup\sqcup\sqcup 01 \\ \to_M \; & \sqcup z_4 \sqcup\sqcup\sqcup\sqcup 01 \\ \to_M \; & \sqcup z_4 \sqcup\sqcup\sqcup\sqcup\sqcup 01 \\ & \vdots \end{aligned}$$

M bleibt im Zustand z_4 und bewegt den Lese-/Schreibkopf immer weiter nach links in den leeren Bereich des Arbeitsbandes.

d) Die von M akzeptierte Sprache ist:

$$L(M) = \{10x \mid x \in \{0,1\}^*\}.$$

Aufgabe 3.

a) Arbeitsweise des Algorithmus:

(1) Überprüfe, ob das Arbeitsband leer ist. Falls ja, dann verwerfe.

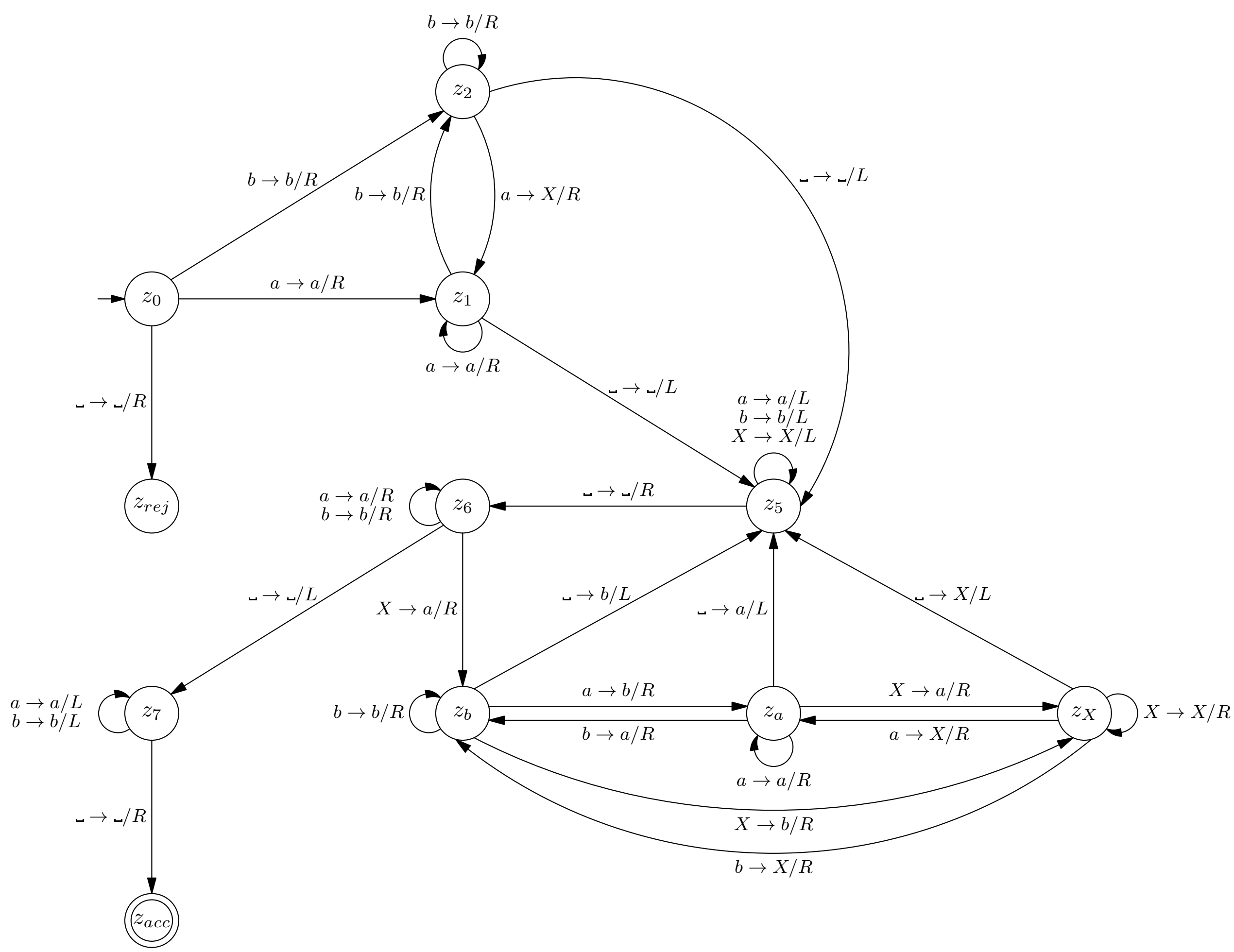
(2) Scanne das Arbeitsband von links nach rechts und suche nach dem Pattern ba. Ersetze jedes Vorkommen von ba durch bX. Wird das Ende der Eingabe erreicht, dann bewege den L/S-Kopf wieder an den Anfang der Eingabe.

(3) Scanne das Arbeitsband von links nach rechts und suche ein X. Wird kein X gefunden, dann springe zu Schritt (5). Ansonsten ersetze das X durch ab. Hierzu muss der Inhalt des Arbeitsbands rechts von X um eine Position nach rechts verschoben werden, um Platz für den zusätzlichen Buchstaben b zu schaffen.

(4) Bewege den L/S-Kopf an den Anfang der Eingabe und wiederhole Schritt 3.

(5) Bewege den L/S-Kopf an den Anfang der Eingabe und akzeptiere.

b) Zustandsdiagramm:

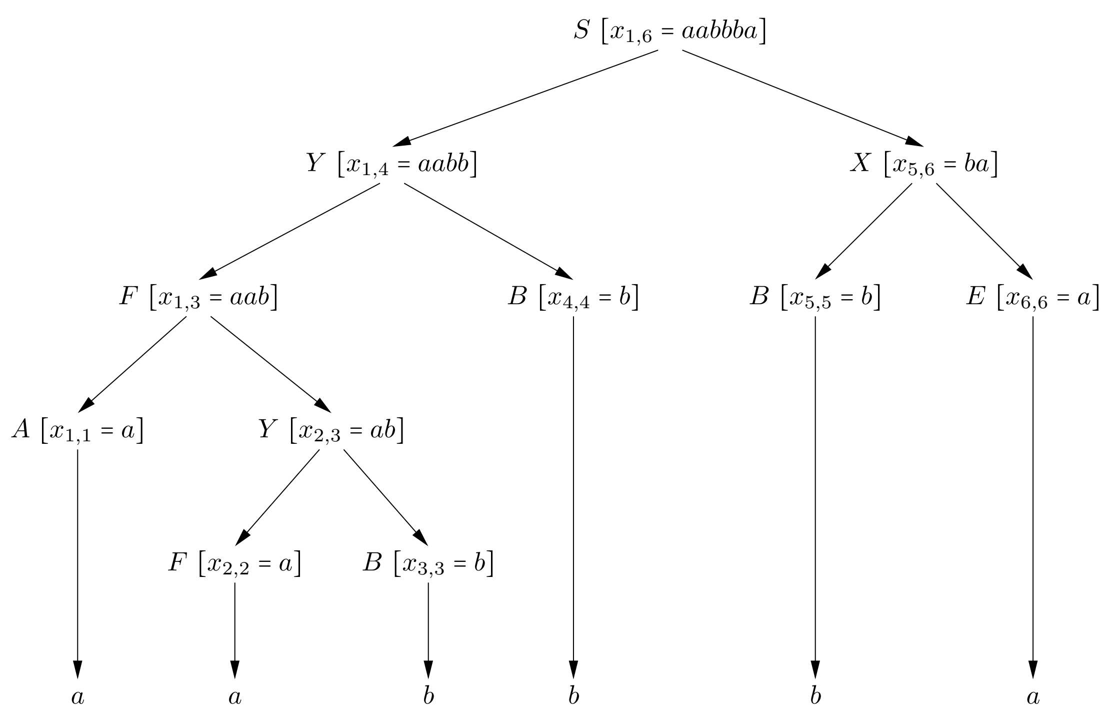


Aufgabe 4.

a) Ergebnis des CYK-Algorithmus für das Wort $aabbba$:

i	1	2	3	4	5	6
	a	a	b	b	b	a
$j=1$	$[E \to a, 0]$ $[F \to a, 0]$ $[A \to a, 0]$	$[E \to a, 0]$ $[F \to a, 0]$ $[A \to a, 0]$	$[B \to b, 0]$	$[B \to b, 0]$	$[B \to b, 0]$	$[E \to a, 0]$ $[F \to a, 0]$ $[A \to a, 0]$
$j=2$		$[Y \to FB, 2]$			$[X \to BE, 5]$	
$j=3$	$[F \to AY, 1]$					
$j=4$	$[Y \to FB, 3]$					
$j=5$						
$j=6$	$[S \to YX, 4]$					

b) Ableitungsbaum für das Wort $aabbba$:



Aufgabe 5. Da jede endliche Sprache regulär ist, sind alle Sprachen in $\mathcal{C}$ entscheidbar und somit auch semi-entscheidbar. Die Sprache $L = \{x0 \mid x \in \{0,1\}^*\}$ enthält unendlich viele Wörter und ist deshalb nicht in $\mathcal{C}$. L ist regulär und somit auch semi-entscheidbar. Folglich ist $\mathcal{C}$ eine nicht-triviale Klasse von semi-entscheidbaren Sprachen. Mit dem Satz von Rice folgt, dass $\mathcal{C}$ nicht entscheidbar ist.

Aufgabe 6.

a) Um zu beweisen, dass $B \in \mathrm{NP}$, wird ein Polynomialzeit Verifikationsalgorithmus konstruiert:

CheckB$(F(x_1,\ldots,x_n), B_1, B_2)$
Input: Aussagenlogische Formel F in 3-KNF, zwei Belegungen B_1 und B_2.
Output: `true` falls B_1 und B_2 zwei verschiedene erfüllende Belegungen für F sind, `false`, sonst.

```
equal := true;
for i := 1 to n do
   if B1[xi] ≠ B2[xi] then
      equal := false;
if equal = true then
   return false
if Evaluate(F, B1) = true and Evaluate(F, B2) = true then
   return true
      else
   return false
```

Die Funktion Evaluate(F, B) steht hierbei für die Implementierung des Circuit-Value-Problems (CVK).

Der Algorithmus gibt genau dann `true` zurück, wenn B_1 und B_2 zwei verschiedene erfüllende Belegungen für F sind.

Die Laufzeit des Algorithmus setzt sich zusammen aus:

- der Überprüfung, ob B_1 und B_2 zwei verschiedene Belegungen sind, und
- dem Test, ob ob B_1 und B_2 erfüllende Belegungen sind.

Sind die Belegungen in einer verketteten Liste gespeichert, dann ist der erste Punkt in einer Laufzeit von $O(n^2)$ durchführbar. Zum Test der Erfüllbarkeit kommt der CVP-Algorithmus zum Einsatz, der in Polynomialzeit in der Länge von F läuft. Insgesamt hat CheckB eine polynomiale Laufzeit in der Länge von F.

Ergebnis: $B \in \mathrm{NP}$.

b) Um zu zeigen, dass B NP-hart ist, reduziert man 3-SAT auf B. Die Reduktionsfunktion f ist definiert als

$$f(F) = \underbrace{(x \vee x \vee \neg x) \wedge F}_{=G},$$

wobei F eine aussagenlogische Formel in 3-KNF und x eine Variable ist, die nicht in F vorkommt. Es ist klar, dass f in Polynomialzeit berechenbar ist.

Es gilt:

- Ist F erfüllbar, dann ist auch G erfüllbar. Da $(x \vee x \vee \neg x)$ eine Tautologie und x eine neue Variable ist, hat G doppelt so viele erfüllende Belegungen wie F, also mindestens 2 erfüllende Belegungen. Folglich ist $G \in B$.
- Angenommen G ist erfüllbar. Da G eine Formel in 3-KNF ist und x in F nicht vorkommt, muss auch F erfüllbar sein. Also ist $F \in$ 3-SAT.

Hieraus folgt, dass

$$F \in \text{3-SAT} \iff f(F) \in B$$

für alle Formeln F in 3-KNF gilt. Also ist f eine Reduktion von 3-SAT auf B und somit B NP-hart.

Aufgabe 7. Gegeben ist die Formel

$$\underbrace{(\neg x_1 \vee x_3 \vee x_4)}_{K_1} \wedge \underbrace{(\neg x_2 \vee x_3 \vee \neg x_1)}_{K_2} \wedge \underbrace{(x_1 \vee \neg x_3 \vee \neg x_4)}_{K_3} \wedge \underbrace{(x_1 \vee \neg x_2 \vee x_4)}_{K_4}.$$

Die Reduktion liefert folgendes Subset Sum Problem:

Zahl	*Variablen*				*Klauseln*			
y_1	1	0	0	0	0	0	1	1
z_1	1	0	0	0	1	1	0	0
y_2	0	1	0	0	0	0	0	0
z_2	0	1	0	0	0	1	0	1
y_3	0	0	1	0	1	1	0	0
z_3	0	0	1	0	0	0	1	0
y_4	0	0	0	1	1	0	0	1
z_4	0	0	0	1	0	0	1	0
g_1	0	0	0	0	1	0	0	0
h_1	0	0	0	0	1	0	0	0
g_2	0	0	0	0	0	1	0	0
h_2	0	0	0	0	0	1	0	0
g_3	0	0	0	0	0	0	1	0
h_3	0	0	0	0	0	0	1	0
g_4	0	0	0	0	0	0	0	1
h_4	0	0	0	0	0	0	0	1
b	1	1	1	1	3	3	3	3

Die Belegung $x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1$ ist eine erfüllende Belegung für diese Formel. Die entsprechende Lösung für Subset Sum ist:

Zahl	*Wert*							
y_1	1	0	0	0	0	0	1	1
z_2	0	1	0	0	0	1	0	1
z_3	0	0	1	0	0	0	1	0
y_4	0	0	0	1	1	0	0	1
g_1	0	0	0	0	1	0	0	0
h_1	0	0	0	0	1	0	0	0
g_2	0	0	0	0	0	1	0	0
h_2	0	0	0	0	0	1	0	0
g_3	0	0	0	0	0	0	1	0
$\sum$	1	1	1	1	3	3	3	3