Prof. Dr. Christoph Karg
Studiengang Informatik
Hochschule Aalen

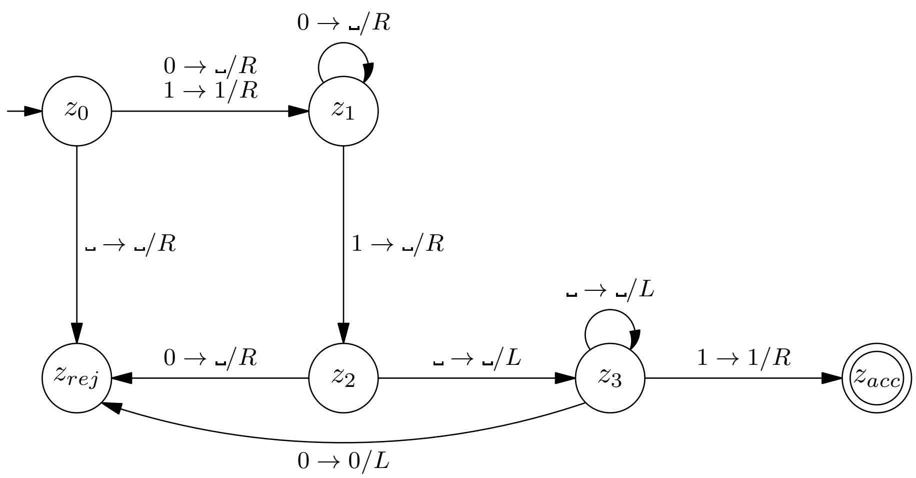
Klausur Berechenbarkeit & Komplexität (Wintersemester 2012/2013)

Lösungshinweise

(Alle Angaben ohne Gewähr[1])

Aufgabe 1.

a) Zustandsdiagramm der Turing Maschine M:



b) Konfigurationsfolge für die Eingabe $x = 10001$:

$$
\begin{array}{rl}
 & \sqcup z_0 10001 \\
\Rightarrow_M & \sqcup 1 z_1 0001 \\
\Rightarrow_M & \sqcup 1 \sqcup z_1 001 \\
\Rightarrow_M & \sqcup 1 \sqcup\sqcup z_1 01 \\
\Rightarrow_M & \sqcup 1 \sqcup\sqcup\sqcup z_1 1 \\
\Rightarrow_M & \sqcup 1 \sqcup\sqcup\sqcup\sqcup z_2 \sqcup \\
\Rightarrow_M & \sqcup 1 \sqcup\sqcup\sqcup z_3 \sqcup\sqcup \\
\Rightarrow_M & \sqcup 1 \sqcup\sqcup z_3 \sqcup\sqcup\sqcup \\
\Rightarrow_M & \sqcup 1 \sqcup z_3 \sqcup\sqcup\sqcup\sqcup \\
\Rightarrow_M & \sqcup 1 z_3 \sqcup\sqcup\sqcup\sqcup\sqcup \\
\Rightarrow_M & \sqcup z_3 1 \sqcup\sqcup\sqcup\sqcup\sqcup \\
\Rightarrow_M & \sqcup 1 z_{acc} \sqcup\sqcup\sqcup\sqcup\sqcup
\end{array}
$$

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

c) Die von M akzeptierte Sprache ist $L(M) = L(10^*1)$.

d) Für die Eingabe $x = 01$ stoppt die Turing Maschine nicht. Betrachte die Konfigurationsfolge, die M auf Eingabe 01 durchläuft:

$$\begin{aligned} & \sqcup z_0 01 \\ \Rightarrow_M \quad & \sqcup\sqcup z_1 1 \\ \Rightarrow_M \quad & \sqcup\sqcup\sqcup z_2 \sqcup \\ \Rightarrow_M \quad & \sqcup\sqcup z_3 \sqcup\sqcup \\ \Rightarrow_M \quad & \sqcup z_3 \sqcup\sqcup\sqcup \\ \Rightarrow_M \quad & \dots \end{aligned}$$

Ab diesem Zeitpunkt bleibt die Turing Maschine im Zustand z_3 und bewegt den L/S-Kopf nach links.

Aufgabe 2. Auswertung der Formel

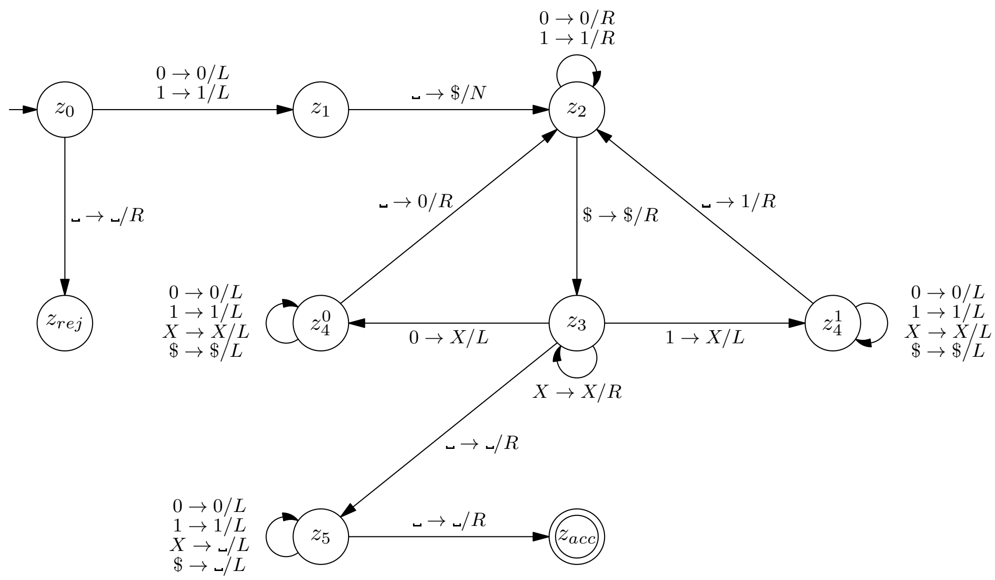
$$F = (x_1 \lor \neg x_3) \land (\neg x_1 \lor x_2) \land (\neg x_3 \lor \neg x_2) \land (\neg x_1 \lor x_3) \land (\neg x_2 \lor x_1)$$

mit dem 2-SAT Algorithmus:

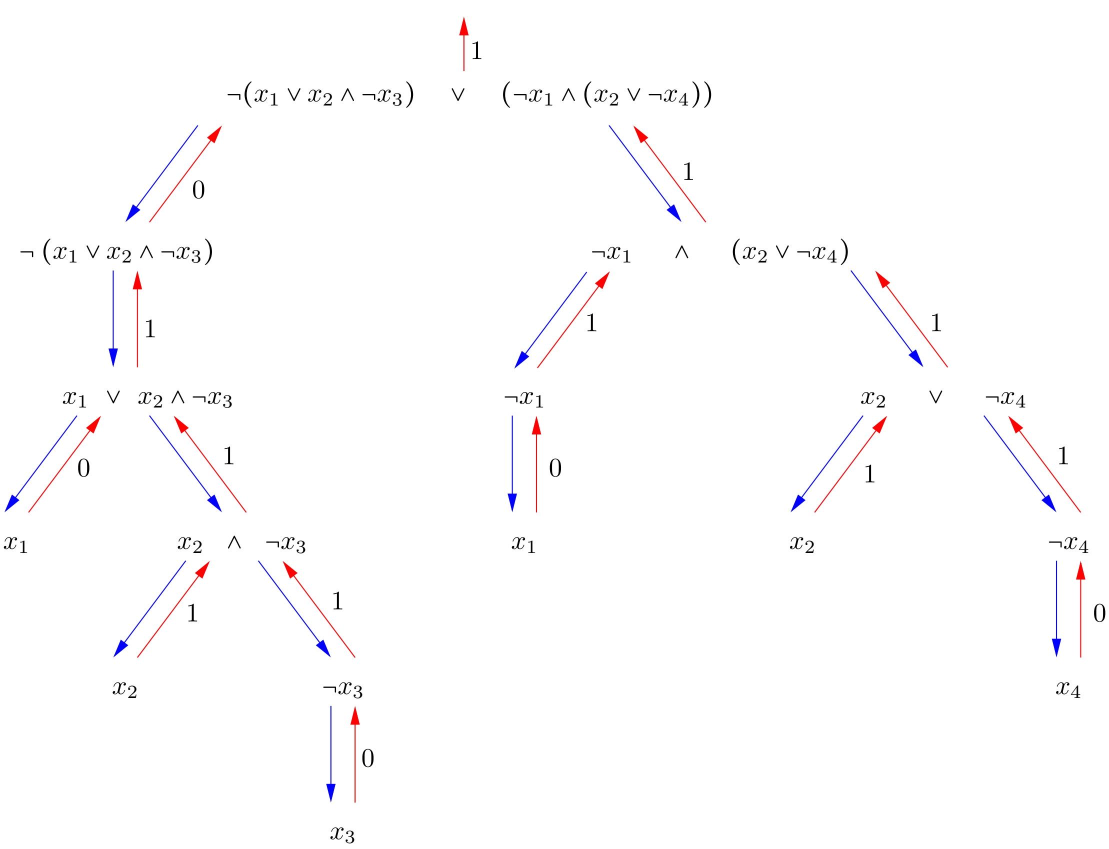
first	x_1	x_2	x_3	$x_1 \lor \neg x_3$	$\neg x_1 \lor x_2$	$\neg x_3 \lor \neg x_2$	$\neg x_1 \lor x_3$	$\neg x_2 \lor x_1$
true	1			✓	$x_2 \to 1$			
true	1	1		✓	✓	$x_3 \to 0$		
true	1	1	0	✓	✓	✓	↯	
false	0			$x_3 \to 0$				
false	0		0	✓	✓	✓	✓	$x_2 \to 0$
false	0	0	0	✓	✓	✓	✓	✓
false	(0)	(0)	(0)	(✓)	(✓)	(✓)	(✓)	(✓)

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

Aufgabe 3. Zustandsdiagramm der Turing Maschine M:



Aufgabe 4. Ablauf des Auswertung für die Belegung $x_1 = 0$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$:



Aufgabe 5. Falls A entscheidbar ist, dann ist auch *Embedding*(A) entscheidbar. Betrachte hierzu folgenden Algorithmus:

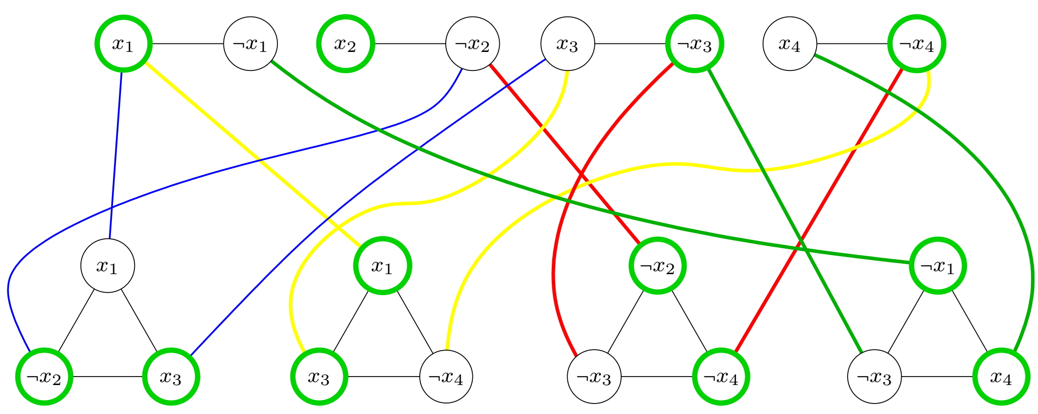
EMBEDDINGA(x)
Input: Wort $x = a_1 a_2 \dots a_n$ der Länge n
Output: `true`, falls $x \in$ *Embedding*(A), `false`, sonst.

```
if x = ε then
   if ε ∈ A then
      return true
   else
      return false
else
   for ℓ := 1 to n do
      for s := 1 to n − ℓ + 1 do
         w := x[s, s + ℓ − 1]
         if w ∈ A then
            return true
   return false
```

Aufgabe 6. Reduktion der Formel

$$F = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee \neg x_4 \vee x_3) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4).$$

auf eine Problemstellung für Vertex Cover, wobei $k = 12$:



Die erfüllende Belegung $x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0$ korrespondiert zu der in den Graphen eingezeichneten Lösung für Vertex Cover.

Aufgabe 7.

a) Die Problemstellung besitzt eine exakte Überdeckung:

$$X_1 = \{a, c\}, X_2 = \{b, e, g\}, X_3 = \{d, f\}$$

b) Betrachte folgenden Verifikationsalgorithmus für *Exact Cover:*

```
CheckExactCover(S, M, X_1, ..., X_k)
Input: Endliche Menge S, Menge M ∈ P(S), Mengen X_1, ..., X_k
Output: true falls X_1, ..., X_k eine exakte Überdeckung von S sind, false, sonst.
1  n := ||S||
2  if k < 1 or k > n then
3     return false
4  for i := 1 to k do
5     if X_i ∉ M then
6        return false
7  for i := 1 to k − 1 do
8     for j := i + 1 to k do
9        for jedes x ∈ X_i do
10          for jedes y ∈ X_j do
11             if x = y then
12                return false
13 Berechne X = X_1 ∪ ... ∪ X_k
14 for jedes x ∈ S do
15    if x ∉ X then
16       return false
17 return true
```

Der Algorithmus akzeptiert genau dann, wenn $X_1, \ldots, X_k$ eine exakte Überdeckung von S ist.

Bei der Laufzeitanalyse wird angenommen, dass die Mengen in einfach verketteten Listen gespeichert wurden, die keine doppelten Elemente enthalten. Die Laufzeit setzt sich aus folgenden Teilen zusammen:

- Berechnung von n (Zeile 1): $O(n)$
- Überprüfung von k (Zeilen 2 und 3): $O(n)$
- Überprüfung, dass jede Menge X_i in M enthalten ist (Zeilen 4 bis 6): $O(n^3)$
- Überprüfung, dass die Mengen X_i und X_j alle paarweise diskunkt sind (Zeilen 7 bis 12): $O(n^4)$

- Berechnung von $X = X_1 \cup \ldots \cup X_k$ (Zeile 13): $O(n^2)$
- Überprüfung, dass $S = X$ (Zeilen 14 bis 16): $O(n^2)$

Laufzeit insgesamt: $O(n^4)$, also polynomial. Somit ist gezeigt, dass *Exact Cover* in NP ist.

Aufgabe 8.

a) Reduktion der Formel

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

auf eine Problemstellung von Subset Sum:

	x_1	x_2	x_3	x_4	K_1	K_2	K_3
y_1	1	0	0	0	1	0	0
z_1	1	0	0	0	0	1	0
y_2	0	1	0	0	1	1	0
z_2	0	1	0	0	0	0	1
y_3	0	0	1	0	0	1	1
z_3	0	0	1	0	0	0	0
y_4	0	0	0	1	1	0	0
z_4	0	0	0	1	0	0	1
g_1	0	0	0	0	1	0	0
h_1	0	0	0	0	1	0	0
g_2	0	0	0	0	0	1	0
h_2	0	0	0	0	0	1	0
g_3	0	0	0	0	0	0	1
h_3	0	0	0	0	0	0	1
b	1	1	1	1	3	3	3

b) Erfüllende Belegung: $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 0$ und die entsprechende Lösung für Subset Sum:

	x_1	x_2	x_3	x_4	K_1	K_2	K_3
y_1	1	0	0	0	1	0	0
z_2	0	1	0	0	0	0	1
y_3	0	0	1	0	0	1	1
z_4	0	0	0	1	0	0	1
g_1	0	0	0	0	1	0	0
h_1	0	0	0	0	1	0	0
g_2	0	0	0	0	0	1	0
h_2	0	0	0	0	0	1	0
Σ	1	1	1	1	3	3	3