Marco Kunze (makunze@cs.tu-berlin.de) **WS 2001/2002**
Sebastian Nowozin (nowozin@cs.tu-berlin.de) **4. 2. 2002**

Lösungen zur 4. Projektaufgabe TheGI1

1. Kodierung für Schaltkreise

Wir definieren einen Kode, der Schaltkreise kodiert. Dazu benötigen wir das Eingabealphabet $\Sigma = \{0, 1, \ldots, 9, \sqcup, \text{“,”}, \$, L, H, \oplus, \otimes, \oslash\}$, wobei 0 bis 9 in bekannter Weise zur Darstellung von Nummern verwendet werden. L und H werden für “*low*” und “*high*” Gatter benutzt, dass heisst für Gatter, die nur 0 beziehungsweise 1 ausgeben. $\sqcup$ und \$ sind Trennzeichen für Gatter beziehungsweise Spalten. Ein Kommata “,” ist das Trennzeichen für Gatternummern. $\oplus$, $\otimes$ und $\oslash$ stehen für die logischen Operationen “*oder*”, “*und*” und “*nicht*”.

Der Schaltkreis wird zur Kodierung in Spalten eingeteilt, so dass die Gatter in jeder Spalte nicht voneinander abhängig sind. Die Spalten werden von links nach rechts auf das Band kodiert, dabei ist jede Spalte durch ein “\$” Zeichen von den benachbarten Spalten getrennt.

Innerhalb jeder Spalte sind die enthaltenen Gatter kodiert. Da die Gatter voneinander unabhängig sind, spielt die Reihenfolge keine Rolle. Zur Vereinfachung - auch der später zu konstruierenden Maschine - nehmen wir an, die Gatter sind fortlaufend kodiert. Die Gatter innerhalb der Spalte sind durch ein “$\sqcup$” voneinander abgetrennt. Die Nummern aller kodierten Gatter sind sequentiell aufsteigend und enthalten keine Lücken.

Jedes Gatter ist so kodiert:

$$\langle\text{Gatternummer}\rangle\langle\text{Operator}\rangle\langle\text{Gatter 1, Gatter 2, } \ldots\rangle$$

Die auf den Operator folgenden Gatternummern geben an, in welche Eingänge der Ausgang dieses Gatters führt.

Für den Schaltkreis der Aufgabenstellung ergibt sich folgende Bandkodierung:

$$1H4\sqcup 2H4,5\sqcup 3L5\$4\otimes 6\sqcup 5\oplus 6,7\$6\otimes 8\sqcup 7\oslash 8\$8\oplus\square$$

Jedes Gatter hat mindestens ein Ausgangsgatter. Eine Ausnahme bildet das Gatter der letzten Spalte. Es besitzt keinen Ausgang, sondern liefert das Resultat des Schaltkreises.

Die Länge eines kodierten Schaltkreises ist polynomiell. Dies lässt sich durch Annahme des ungünstigsten Falls zeigen. Da wir die Zustandsnummern auf dem Band als Dezimalzahlen kodieren, ist die maximale Länge einer solchen Nummer durch den Logarithmus zur Basis 10 gegeben. Wir definieren eine die maximale Länge einer auf dem Band befindlichen Zustandsnummer durch die Anzahl der Zustände:

$$z_m := \lfloor \log(n) \rfloor + 1$$

Ein einzelnes Gatter mit anschliessendem Trennzeichen hat dann die Länge:

$$|\text{Gatter}| = \underbrace{z_m}_{\text{Zust.nr}} + \underbrace{1}_{\text{Op}} + \underbrace{(n-1)}_{|\text{Ausgänge}|} \cdot \underbrace{(z_m+1)}_{\text{Nr“,”}} + \underbrace{1}_{\sqcup} - \underbrace{1}_{\text{letztes “,”}}$$

Damit folgt für die Gesamtlänge bei n Zuständen:

$$\begin{aligned}
|\text{Schaltkreis}| &\leq n \cdot (z_m + (z_m+1)(n-1) + 1) \\
&= n(\lfloor \log(n) \rfloor + 1) + n(n-1)(\lfloor \log(n) \rfloor + 2) + n \\
&= n + n\lfloor \log(n) \rfloor + (n^2 - n)(\lfloor \log(n) \rfloor + 2) + n \\
&= n^2 \lfloor \log(n) \rfloor + 2n^2 - n + n \qquad (1) \\
&< n^3 + 2n^2
\end{aligned}$$

Im Schritt (1) nutzen wir, dass für alle $n \in \mathbb{N}$, $n \geq 1$ gilt: $n > \lfloor \log(n) \rfloor$.

2. DTM: Entscheider für $w \in CVP$

Sei im folgenden M die Maschine, die $w \in CVP$ entscheidet. Wir beschreiben die Funktionsweise von M. Eine sehr abstrahierte Darstellung von M ist in Abbildung 1 gezeigt. $L(M)$ ist die Menge aller korrekten Schaltkreise, die nach den definierten Regeln kodiert worden sind und bei der Auswertung Strom auf dem letzten Gatterausgang haben.

Falls M akzeptierend terminiert, ist $w \in CVP$. Terminiert M nicht akzeptierend, so ist $w \notin CVP$.

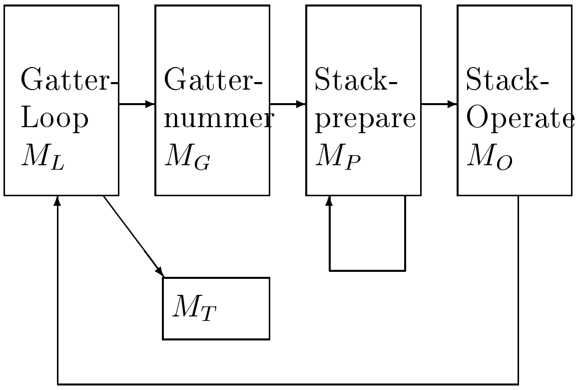

Abbildung 1: M: Symbolischer CVP Entscheider

- M besitzt vier Bänder
  - das Eingabeband mit dem kodierten Schaltkreis
  - das Resultatband, das für jedes Gatter in der Ordnung des Eingabebandes die Ausgangszustände des Gatters enthält
  - das Zustandsband, das beim Auswerten zum Speichern des aktuellen Zustandes verwendet wird
  - das Stackband (oder auch Kellerband), das die Parameter für die Operatoren liefert
- M läuft in einem äusseren Durchlauf durch M_L von links nach rechts die Gatter des Eingabebandes ab, und füllt dabei jeweils ein Resultatwert auf dem Ausgabeband auf
- M_G, M_P und M_O werden für jedes Gatter durchlaufen:
  - in M_G, der "Gatternummermaschine", wird die Gatternummer auf das Zustandsband kopiert, und dabei überprüft, ob die Gatternummer genau eins grösser ist als die vorherige. Ist dies nicht der Fall, so wird nicht akzeptierend terminiert.
  - in M_P, der "Preparationsmaschine", wird das Eingabeband zuerst zur vorherigen Spalte zurückgespult, dann gatterweise nach links durchlaufen. Dabei geht pro Gatter der Resultatband-Kopf eins nach links. Zeigt ein Ausgang eines Gatters auf das aktuelle Gatter (vgl. mit Gatternummer auf Zustandsband), wird das gespeicherte Ergebnis des Gatters vom Resultatband auf das Stackband kopiert
  - immer noch in M_P, wird bei Erreichen des linken Endes des Schaltkreiskodes zum aktuellen Gatter vorgespult (machbar durch Resultatband), dort der Lesekopf auf den Operator gesetzt
  - in M_O, der "Operatormaschine" wird der Operator des aktuellen Gatters dekodiert und die Werte auf dem Stackband dementsprechend verknüpft. Das Resultat wird auf dem Resultatband am rechten Ende gespeichert. In der folgenden Tabelle ist s_0 das oberste Stackelement, s_1 das darunterliegende, usf.

$w = \lvert \text{Stack} \rvert$	Verknüpfung	Resultat
$w = 0$	L	0
$w = 0$	H	1
$w = 1$	$\oslash$	not s_0
$w = 2$	$\oplus$	s_0 or s_1
$w = 2$	$\otimes$	s_0 and s_1

  * es werden w Elemente vom Stapel genommen, der Stapel muss genau w Elemente enthalten (Check durch "mindestens w für Operation" und "nach Operation muss Stack leer sein").
  * ein falscher Schaltkreiskode ist dann vorhanden wenn nicht genau w Elemente gespeichert wurden. Es wird nicht akzeptierend terminiert.
  * dadurch wird garantiert, das pro Gatter genau ein Resultatwert auf dem Resultatband gespeichert wird. Das Eingabe- und Resultatband sind auf die Gatter bezogen synchron.

- wird beim Erreichen des Gatterendes ein Blank ("$\Box$") in M_L gelesen, so wird nach M_T überführt.

- M_T ist die "Terminationsmaschine". Sie vergleicht die aktuelle Gatternummer mit allen Ausgangsgatternummern. Dazu geht der Lesekopf auf dem Eingabeband nach links und liest dabei jede Ausgangsgatternummer jedes Gatters. Wenn die Ausgangsnummer grösser als die aktuelle Gatternummer (im Zustandsband gespeichert) ist, so wird nicht akzeptierend terminiert. Wird am linken Ende des Eingabebandes ein "$\Box$" gelesen so ist der Schaltkreis syntaktisch korrekt. Dann wird überprüft ob der letzte Resultatwert positiv war. Ist dies der Fall, so liegt auf dem letzten Gatterausgang ein high Pegel an und es wird akzeptierend terminiert, da der Schaltkreis $\in CVP$ ist. Andernfalls wird nicht akzeptierend terminiert.

Der Schaltkreiskode wird syntaktisch auf Korrektheit geprüft. Dies geschieht durch folgende Elemente in M:

- durch Vergleich der Gatternummer mit einem eigenen Zähler in M_G wird gewährleistet dass keine Gatternummern doppelt verwendet werden oder die Gatter in der falschen Reihenfolge kodiert sind

- durch Überprüfung der Anzahl der Stack Elemente in M_O wird gewährleistet, dass genau die richtige Anzahl der Eingänge am Gatter vorhanden sind

- dadurch, dass in M_P nur Spalten links von der aktuellen betrachtet werden, können sich die Eingänge des aktuellen Gattern nur auf bereits berechnete Gatter beziehen oder falsch sein, was in M_O erkannt wird.

- durch die Überprüfung in M_T wird gewährleistet, dass kein Ausgang eines Gatters "ins Leere" zeigen kann, und dass zuletzt berechnete Gatter wirklich der Ausgang des Schaltkreises ist.

3. $CVP \in P$

Wenn $CVP \in P$ ist, dann existiert ein Polynom p, so dass für alle Eingaben w eine obere Zeitschranke $p(|w|)$ gesetzt ist.

Da M für alle Eingaben terminiert gilt:

$$T_M(n) = \max \{t_M(w) \mid |w| \leq n\}$$

Es ist für alle $n \in \mathbb{N}$ zu zeigen:

$$p(n) \geq T_M(n)$$

Behauptung: Ein solches Polynom existiert.

Beweis: Idee: Wir nehmen für jede Teilmaschine in M den jeweils ungünstigsten Fall an. Dann berechnen wir für jede Teilmaschine die maximale Schrittanzahl. Wir zeigen, dass sich $T_M(n)$ leicht aus den Zeitaufwänden der Untermaschinen errechnen lässt. Ist selbst im ungünstigsten Fall (*worst case*) der Zeitaufwand für T_M polynom, so ist er dies für alle anderen Fälle auch und die Behauptung ist bewiesen.

Zur leichteren Darstellung definieren wir einige Hilfsvariablen

- Länge des Eingabewortes von M: $|w| = n$
- Kleinste mögliche kodierte Gatterlänge: $g_{min} = 3$
- Maximale Anzahl kodierter Gatter auf dem Eingabeband: $z = \frac{|w|}{g_{min}} + 1 = \frac{|w|}{3} + 1$
- Maximale Länge einer kodierten Gatternummer: $l = \lfloor \log(z) \rfloor + 1$

Wir berechnen die Aufwände für den jeweils ungünstigsten Fall in den Teilmaschinen von M

- M_L: läuft bei jedem Schritt nach rechts, bis das Ende des Eingabewortes erreicht ist oder ein Gatter gefunden wurde
  $\Rightarrow$ maximal $|w|$ Schritte
- M_G: kopiert und vergleicht die aktuelle Gatternummer zusammen in maximal l Schritten
  $\Rightarrow$ maximal l Schritte
- M_P:
  - spult einmalig bis zur vorherigen Spalte, maximal $|w|$ Schritte
  - maximal z mal wird beim Spulen nach links ein Gatter gesehen, für jedes Gatter ist der Aufwand im ungünstigsten Fall $(l + l + 1)$:
    * Vergleich einer Gatternummer mit der auf dem Zustandsband gespeicherten Nummer: maximal l Schritte
    * Zurückspulen des Zustandsbandes vor jedem Vergleich: maximal l Schritte
  - spult einmalig bis zum aktuellem Gatter nach rechts, maximal $|w|$ Schritte

  $\Rightarrow$ maximal $|w| + z\,(l + l + 1) + |w|$ Schritte
- M_O:
  - Operator dekodieren: 1 Schritt
  - Verknüpfung und dabei Stackelemente löschen: maximal 2 Schritte
  - Speichern des Resultates: 1 Schritt
  - "Stackband leer" Überprüfung: 1 Schritt

  $\Rightarrow$ maximal $1 + 2 + 1 + 1 = 5$ Schritte
- M_T: Vergleiche der Gatternummern und jeweils Zurückspulen des Zustandsbandes: maximal $|w| + z \cdot l$ Schritte. Die Termination selber benötigt keinen Schritt.
  $\Rightarrow$ maximal $|w| + z \cdot l$ Schritte

T_M setzt sich aus zwei Teilen zusammen: $T_{M_L}, T_{M_G}, T_{M_P}$ und T_{M_O} werden für jedes Gatter durchlaufen, T_{M_T} jedoch höchstens einmal. Daher ergibt sich für den Aufwand T_M im ungünstigsten Fall:

$$
\begin{aligned}
T_M(n) &\leq z \cdot \Big[\underbrace{T_{M_L}(|w|) + T_{M_G}(|w|) + T_{M_P}(|w|) + T_{M_O}(|w|)}_{\text{wird für jedes kodierte Gatter durchlaufen}}\Big] + T_{M_T}(|w|) \\
&= z \cdot \Big[|w| + l + |w| + z \cdot (l + l + 1) + |w| + 5\Big] + |w| + z \cdot l \\
&= z \cdot \Big[3\,|w| + 2zl + z + l + 5\Big] + |w| + zl \\
&= \left(\frac{|w|}{3} + 1\right) \cdot \left[3\,|w| + 2\left(\frac{|w|}{3} + 1\right) l + \frac{|w|}{3} + 1 + l + 5\right] + |w| + \left(\frac{|w|}{3} + 1\right) l \qquad (2) \\
&< \left(\frac{n}{3} + 1\right) \cdot \left[3n + \left(\frac{2n}{3} + 2\right) n + \frac{n}{3} + n + 6\right] + n + \frac{n^2}{3} + n \qquad (3) \\
&= \frac{n}{3}\left(\frac{2}{3}n^2 + \frac{19}{3}n + 6\right) + n^2 + \frac{25}{3}n + 6 \\
&= \frac{2}{9}n^3 + \frac{28}{9}n^2 + \frac{31}{3}n + 6 \\
&= p(n)
\end{aligned}
$$

Im Schritt von (2) zu (3) nutzen wir, dass $\forall n \in \mathbb{N} \setminus \{0\}$: $n > \lfloor \log\left(\frac{n}{3} + 1\right) \rfloor = l$ ist.

Da wir für den ungünstigsten Fall der Maschine zeigen konnten, dass das Polynom $p(n) = \frac{2}{9}n^3 + \frac{28}{9}n^2 + \frac{31}{3}n + 6$ für jedes Eingabewort w mit $n = |w|$ den Aufwand $T_M(n)$ begrenzt, begrenzt es auch $T_M(n)$ für alle anderen Fälle. Damit ist $CVP \in P$ bewiesen.