Computation Tree Logic

Syntax

Wir betrachten die temporale Aussagenlogik *CTL*,
(*Computation Tree Logic*)

Definition der CTL-Formeln

1. jede aussagenlogische Variable (aus P_0) ist eine CTL-Formel,
2. Sind F, G CTL-Formeln, dann auch die aussagenlogischen Kombinationen davon, d.h., $\neg F$, $F \vee G$, $F \wedge G$, etc.,
3. sind F, G CTL-Formeln, dann sind auch

$$\begin{array}{l} all\ next\ F, \\ ex\ next\ F, \\ all(F\ until\ G), \\ ex(F\ until\ G) \end{array}$$

CTL-Formeln.

Computation Trees

Ein Berechnungsbaum $\mathcal{T} = (T, R, S, v)$ besteht aus

1. einer endlichen Menge S von Zuständen,
2. einem Baum (T, R), dessen Knoten mit Elementen aus S markiert sind,
3. einer Bewertungsfunktion $v : S \times P_0 \rightarrow \{0, 1\}$

Dabei bedeutet $R(n_1, n_2)$, daß der Knoten n_2 ein unmittelbarer Nachfolger von n_1 ist.
Wurzel von (T, R) = n_0.
$s(n) \in S$ = Markierung von n.
Ein Pfad in $\mathcal{T}$ ist eine endliche oder unendliche $n_1, \ldots, n_k, \ldots$ Folge von Knoten, so daß für alle $i > 0$ gilt $R(n_{i-1}, n_i)$.

Semantik

Abkürzung: $n \models F$ anstelle von $(\mathcal{T}, n) \models F$.

1	$n \models p$	gdw	$v(s(n), p) = 1$
2	$n \models \neg F$	gdw	$n \not\models F$
3	$n \models F \wedge G$	gdw	$n \models F$ und $n \models G$
4	$n \models \mathit{all\ next}\ F$	gdw	für alle $n_1 \in T$ mit $R(n, n_1)$ gilt $n_1 \models F$
5	$n \models \mathit{ex\ next}\ F$	gdw	es gibt ein $n_1 \in T$ mit $R(n, n_1)$, so daß $n_1 \models F$ gilt,

Semantik (Forts.)

Abkürzung: $n \models F$ anstelle von $(\mathcal{T}, n) \models F$.

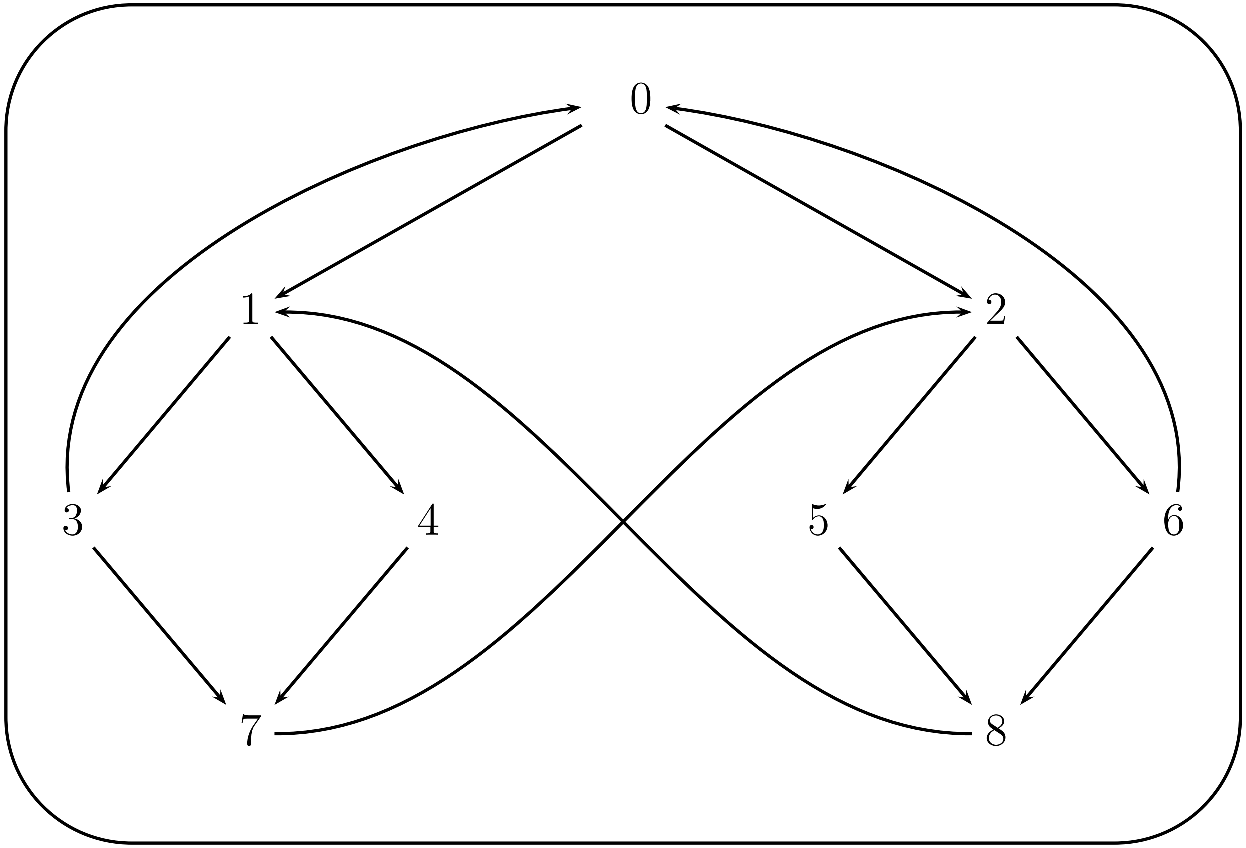
6	$n \models all(F\ until\ G)$	gdw	für jeden Pfad $n = n_1, n_2, \ldots$ gibt es ein $i \geq 1$, so daß $n_i \models G$ und $n_j \models F$ für alle j mit $1 \leq j < i$
7	$n \models ex(F\ until\ G)$	gdw	es gibt einen Pfad $n = n_1, n_2, \ldots$ und ein $i \geq 1$, so daß $n_i \models G$ und $n_j \models F$ für alle j mit $1 \leq j < i$

Von Kripke zu CT

Sei $\mathcal{K} = (G, R, v)$ eine aussagenlogische Kripke Struktur und $s \in G$.

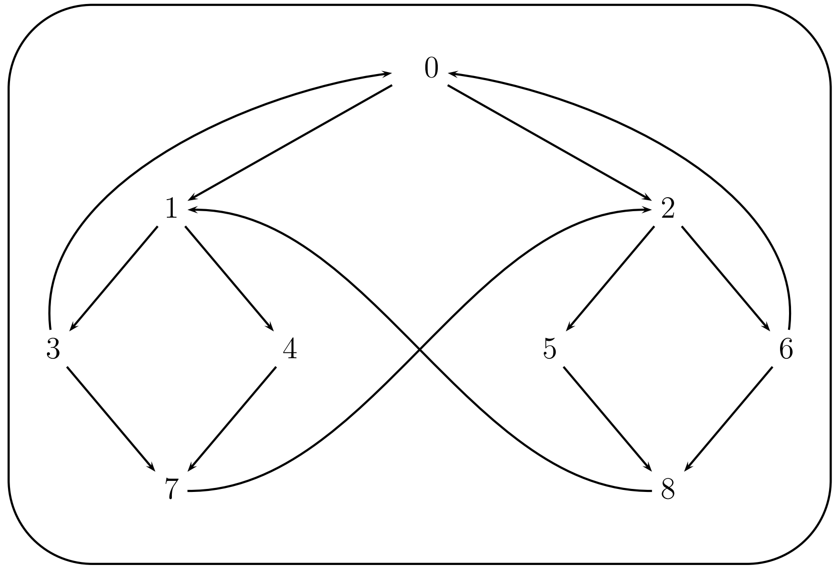
1. Eine Berechnungsfolge für $(\mathcal{K}, s)$ ist eine endliche oder unendliche Folge von Zuständen (Welten) $s_i \in G$, so daß $s_0 = s$ und für alle s_i in der Folge mit $i > 0$ gilt $R(s_{i-1}, s_i)$.
2. Eine Berechnungsfolge heißt maximal, wenn sie unendlich ist oder mit einem Zustand s_n endet, der keinen R-Nachfolger in G besitzt.
3. Der Berechnungsbaum $\mathcal{T} = (T, R, S, v)$ zu $(\mathcal{K}, s)$ entsteht, indem wir $S = G$ setzen, R und v unverändert aus der Kripke Struktur übernehmen und (T, R) so wählen, daß die maximalen Pfade in (T, R) die maximalen Berechnungsfolgen für $(\mathcal{K}, s)$ sind.

Kripke Struktur $\mathcal{K}_{me}$



N_i	Prozeß i befindet sich in einer nichtkritischen Region
T_i	Prozeß i befindet sich in der Anmeldephase
C_i	Prozeß i befindet sich in einer kritischen Region

Kripke Struktur $\mathcal{K}_{me}$



	0	1	2	3	4	5	6	7	8
N_1	1	0	1	0	0	0	1	0	0
N_2	1	1	0	1	0	0	0	0	0
T_1	0	1	0	0	1	1	0	0	1
T_2	0	0	1	0	1	1	0	1	0
C_1	0	0	0	1	0	0	0	1	0
C_2	0	0	0	0	0	0	1	0	1

Berechnungsbaum zu $\mathcal{K}_{me}$