5 Die Logik CTL*

Zur Erinnerung: Die im letzten Kapitel betrachtete Logik LTL wird über Läufen von Transitionssystemen interpretiert. Eine Interpretation über Zuständen ist durch die Konvention, dass über alle aus dem gegebenen Zustand ausgehenden Läufe quantifiziert wird, ebenfalls möglich. Dadurch ist es ebenfalls möglich, LTL bzgl. ihrer Ausdrucksstärke z.B. mit der Baumzeitlogik CTL zu vergleichen.

Genauso hätte man eine existenzielle Laufquantifizierung als Konvention einführen können. Dann hätten sich lediglich einige Komplexitätsresultate geändert: Das Model Checking Problem für $\text{LTL}^{\mathsf{min}}$ wäre dann NP-vollständig. Im Bezug auf den Vergleich zu CTL hätte sich jedoch nichts geändert. Es bietet sich somit an, diese Möglichkeiten explizit in der Logik zuzulassen, d.h. die Quantifizierung über alle oder einen Lauf nicht implizit vorauszusetzen, sondern in der Formel festzuschreiben. Damit kommen wir zu der sogenannten vollen Baumzeitlogik CTL*.

5.1 Syntax und Semantik

Definition 5.1
Sei $\mathcal{P}$ eine Menge von atomaren Propositionen. Formeln der Logik *TL* sind gegeben durch die folgende Grammatik.

$$\varphi \quad ::= \quad q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathtt{X}\varphi \mid \varphi\mathtt{U}\varphi \mid \varphi\mathtt{R}\varphi \mid \mathtt{A}\varphi \mid \mathtt{E}\varphi$$

wobei $q \in \mathcal{P}$.

TL-Formeln werden zuerst einmal über Läufen innerhalb eines Transitionssystems interpretiert. Beachte den Unterschied zu CTL, wo Läufe irrelevant sind, sowie den Unterschied zu LTL, wo das zugrundeliegende Transitionssystem keine Rolle spielt.

Definition 5.2
Sei $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ ein totales, knotenbeschriftetes Transitionssystem und Π_s für jedes $s \in \mathcal{S}$ die Menge aller Läufe in $\mathcal{T}$, die in s beginnen. Sei π ein beliebiger Lauf. Die

Semantik der Logik TL ist induktiv definiert wie folgt.

$$\begin{aligned}
\mathcal{T}, \pi \models q \quad &\text{gdw.} \quad \pi = s \ldots \text{ und } q \in \lambda(s) \\
\mathcal{T}, \pi \models \varphi \vee \psi \quad &\text{gdw.} \quad \mathcal{T}, \pi \models \varphi \text{ oder } \mathcal{T}, \pi \models \psi \\
\mathcal{T}, \pi \models \varphi \wedge \psi \quad &\text{gdw.} \quad \mathcal{T}, \pi \models \varphi \text{ und } \mathcal{T}, \pi \models \psi \\
\mathcal{T}, \pi \models \neg\varphi \quad &\text{gdw.} \quad \mathcal{T}, \pi \not\models \varphi \\
\mathcal{T}, \pi \models \mathtt{X}\varphi \quad &\text{gdw.} \quad \mathcal{T}, \pi^{(1)} \models \varphi \\
\mathcal{T}, \pi \models \varphi \mathtt{U} \psi \quad &\text{gdw.} \quad \exists k \in \mathbb{N} \text{ mit } \mathcal{T}, \pi^{(k)} \models \psi \text{ und } \forall j \leq k : \mathcal{T}, \pi^{(j)} \models \varphi \\
\mathcal{T}, \pi \models \varphi \mathtt{R} \psi \quad &\text{gdw.} \quad \forall k \in \mathbb{N} : \mathcal{T}, \pi^{(k)} \models \psi \text{ oder } \exists j \leq k \text{ mit } \mathcal{T}, \pi^{(j)} \models \varphi \\
\mathcal{T}, \pi \models \mathtt{A}\varphi \quad &\text{gdw.} \quad \pi = s \ldots \text{ und } \forall \pi' \in \Pi_s : \mathcal{T}, \pi' \models \varphi \\
\mathcal{T}, \pi \models \mathtt{E}\varphi \quad &\text{gdw.} \quad \pi = s \ldots \text{ und } \exists \pi' \in \Pi_s : \mathcal{T}, \pi' \models \varphi
\end{aligned}$$

Dabei bezeichnet $\pi^{(i)}$ für ein $i \in \mathbb{N}$ wieder das i-te Suffix von π.

Im folgenden werden Transitionssysteme immer als total angenommen.

Um TL auch über Zuständen von Transitionssystemen zu interpretieren, führen wir das Konzept einer Zustandsformel ein.

Definition 5.3
Eine TL-Formel φ heißt *Zustandsformel*, falls für alle Transitionssysteme $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ und alle Läufe $\pi = s \ldots$ und $\pi' = s \ldots$ gilt:

$$\mathcal{T}, \pi \models \varphi \quad \text{gdw.} \quad \mathcal{T}, \pi' \models \varphi$$

Alle atomaren Propositionen sind z.B. Zustandsformeln, außerdem jede Tautologie. Die Menge der Zustandsformeln ist abgeschlossen unter den booleschen Operatoren $\vee$, $\wedge$ und $\neg$, und jede Formel der Form $\mathtt{A}\psi$ ist offensichtlich eine Zustandsformel. Dasselbe gilt übrigens auch für Formeln der Form $\mathtt{E}\psi$.

Die Logik *CTL** ist die Menge aller Zustandsformeln aus TL. Beachte, dass für eine Zustandsformel φ gilt: $\varphi \equiv \mathtt{A}\varphi$. Da außerdem die rechte Seite dieser Gleichung auf jeden Fall eine Zustandsformel ist, können wir einfach davon ausgehen, dass eine CTL*-Formel von der Form $\mathtt{A}\varphi$ ist, wobei φ eine beliebige TL-Formel ist. Somit lässt sich die Erfüllung einer CTL*-Formel auch eindeutig in Bezug auf einen Zustand angeben: $\mathcal{T}, s \models \varphi$. Beachte aber, dass eine Unterformel einer CTL*-Formel keine Zustandsformel sein muss.

Lemma 5.1
Für alle Zustandsformeln $\varphi, \psi \in$ TL gilt in CTL*:

$$\models \psi \ \wedge \ \mathtt{AG}(\psi \rightarrow \mathtt{EX}(\varphi \mathtt{U} \psi)) \ \rightarrow \ \mathtt{EG}(\varphi \mathtt{U} \psi)$$

Beweis Übung. ∎

5.2 Ausdrucksstärke

Lemma 5.2
In CTL* gelten die folgenden Äquivalenzen:

a) $Q_1Q_2\varphi \equiv Q_2\varphi$ für alle $Q_1Q_2 \in \{\mathtt{E}, \mathtt{A}\}$,

b) $\neg\mathtt{E}\varphi \equiv \mathtt{A}\neg\varphi$.

Beweis Übung. ∎

Wegen (a) muss man bei Formeln, die aufgrund eines top-level Laufquantors bereits offensichtlich Zustandsformeln sind, nicht noch den implizit angenommenen A-Quantor einfügen. Wegen (b) und dem entsprechenden Resultat für LTL kann man sich bei CTL* ebenfalls wieder auf Formeln in positiver Normalform beschränken.

Satz 5.1
Es gilt CTL $\lneq$ CTL* und LTL $\lneq$ CTL*, wobei LTL-Formeln über Zuständen interpretiert werden.

Beweis CTL $\leq$ CTL* gilt trivialerweise, da CTL ein syntaktisches Fragment von TL ist und jede CTL-Formel per Definition bereits eine Zustandsformel ist. Ausserdem gilt LTL $\leq$ CTL*, da auch LTL ein syntaktisches Fragment von TL ist, und für eine LTL-Formel φ offensichtlich gilt: $\mathcal{T}, s \models \varphi$ gdw. $\mathcal{T}, s \models \mathtt{A}\varphi$, wobei $\mathtt{A}\varphi$ dann eine CTL*-Formel ist.

Die Striktheit beider Inklusionen ist eine Konsequenz aus Satz 4.1, welcher besagt, dass CTL und LTL unvergleichbar sind. Angenommen es gelte CTL $\equiv$ CTL*, dann hätten wir auch LTL $\leq$ CTL, was Satz 4.1 widerspricht. Umgekehrt genauso. ∎

Beispiel 5.1
In Satz 3.11 wurde gezeigt, dass man in CTL nicht "es gibt einen Lauf, auf dem unendlich oft q gilt" ausdrücken kann. In CTL* ist dies jedoch ohne weiteres möglich: $\mathtt{EGF}q$.

Beispiel 5.2
Beachte, dass CTL als syntaktisches Fragment von CTL* nicht mit der Menge der CTL*-Formeln übereinstimmt, die eine CTL-definierbare Eigenschaft ausdrücken. So ist z.B. $\mathtt{A}(\mathtt{X}q \vee \mathtt{XX}q) \equiv \mathtt{AX}(q \vee \mathtt{AX}q)$, obwohl der universelle Quantor im allgemeinen nicht mit dem booleschen $\vee$, sondern nur mit $\wedge$ kommutiert.

Lemma 5.3
In CTL* gelten die folgenden Äquivalenzen.

a) $\mathtt{A}(\varphi \wedge \psi) \equiv \mathtt{A}\varphi \wedge \mathtt{A}\psi$,

b) $\mathtt{E}(\varphi \vee \psi) \equiv \mathtt{E}\varphi \vee \mathtt{E}\psi$,

c) $Q\mathtt{X}\varphi \equiv Q\mathtt{X}Q\varphi$ für jedes $Q \in \{\mathtt{A}, \mathtt{E}\}$.

Beweis Übung. ∎

Äquivalenzen für die anderen Kombinationen ergeben sich nur auf eingeschränkten Modellklassen.

Lemma 5.4
Auf linearen Transitionssystemen gelten die Äquivalenzen $\mathtt{A}(\varphi \vee \psi) \equiv \mathtt{A}\varphi \vee \mathtt{A}\psi$ und $\mathtt{E}(\varphi \wedge \psi) \equiv \mathtt{E}\varphi \wedge \mathtt{E}\psi$.

Beweis Übung. ∎

5.3 Komplexität

Definition 5.4
Die Quantorentiefe einer CTL*-Formel φ, kurz $qd(\varphi)$, ist die maximale Anzahl der Operatoren A und E auf einem Pfad des Syntaxbaums von φ.

Satz 5.2
Das Model Checking Problem für CTL* ist PSPACE-vollständig.

Beweis Die untere Schranke ergibt sich sofort aus Satz 5.1 und der PSPACE-Härte des Model Checking Problems für LTL (Satz 4.6). Für die obere Schranke benutzen wir ebenfalls das LTL Model Checking Problem.

Sei ein Transitionssystem $\mathcal{T} = (\mathcal{S}, \rightarrow, \lambda)$ gegeben, wobei $\lambda : \mathcal{S} \rightarrow 2^{\mathcal{P}}$ für ein $\mathcal{P}$, und sei $\varphi \in$ CTL*. Da $\mathtt{E}\psi \equiv \neg\mathtt{A}\neg\psi$ gilt, können wir o.B.d.A. davon ausgehen, dass in φ der Operator E nicht vorkommt. Definiere neue atomare Propositionen $\mathcal{P}_\varphi := \{q_\psi \mid \psi \in Sub(\varphi) \text{ und } \psi = \mathtt{A}\psi' \}$ und $\mathcal{P}' := \mathcal{P} \cup \mathcal{P}_\varphi$.

Um zu entscheiden, ob $\mathcal{T}, s \models \varphi$ für ein bestimmtes $s \in \mathcal{S}$ gilt, gehen wir nun folgendermaßen vor. Seien $\mathtt{A}\psi_1, \ldots, \mathtt{A}\psi_n$ alle Unterformeln von φ, die mit dem universellen Quantor A beginnen, so dass $qd(\psi_i) = 0$ für alle $i = 1, \ldots, n$ gilt. Somit sind alle ψ_i LTL-Formeln, und laut Satz 4.7 kann mit polynomiellem Platz für jedes $i = 1, \ldots, n$ und jedes $t \in \mathcal{S}$ bestimmt werden, ob $\mathcal{T}, t \models \psi_i$ gilt oder nicht. Beachte, dass Platz wiederverwendbar ist.

In einem zweiten Schritt sei $\mathcal{T}' := (\mathcal{S}, \rightarrow, \lambda')$ mit

$$\lambda'(t) \;:=\; \lambda(t) \cup \{q_{\psi_i} \mid i \in \{1, \ldots, n\} \text{ und } \mathcal{T}, t \models \mathtt{A}\psi_i\}$$

Definiere außerdem $\varphi' := \varphi[q_{\psi_1}/\mathtt{A}\psi_1, \ldots, q_{\psi_n}/\mathtt{A}\psi_n]$. Offensichtlich gilt für alle $t \in \mathcal{S}$: $\mathcal{T}, t \models \varphi$ gdw. $\mathcal{T}', t \models \varphi'$. Beachte, dass gilt: $qd(\varphi') < qd(\varphi)$, weswegen sich dieses Verfahren iterieren lässt bis ein φ' mit $qd(\varphi') = 1$ erreicht ist. Da angenommen wird, dass φ selbst von der Form $\mathtt{A}\psi$ ist, ist die Frage, ob $\mathcal{T}, s \models \varphi$ gilt, damit auf die Frage reduziert worden, ob $\mathcal{T}', s \models q_\psi$ gilt, wobei $\mathcal{T}'$ die Erweiterung von $\mathcal{T}$ um die Propositionen $\mathcal{P}_\varphi$ ist.

Beachte, dass durch die jeweilige Wiederverwendung des Platzes, der zum LTL Model Checking benötigt wird, insgesamt der Platzbedarf polynomiell platzbeschränkt ist. Zusätzlich muss lediglich die Erweiterung von λ auf $\mathcal{P}_\varphi$ gespeichert werden. ∎

Satz 5.3
Das Erfüllbarkeitsproblem für CTL* ist 2-EXPTIME-hart.

Beweis Wird analog zu Satz 3.9 (EXPTIME-Härte des Erfüllbarkeitsproblems für CTL) durch Reduktion auf das Wortproblem für alternierende, exponentiell platzbeschränkte Turing Maschinen bewiesen. ■

Da eine Konfiguration solch einer Turing Maschine nicht mehr nur polynomielle Länge hat, wird die Reduktion etwas komplizierter. So lässt sich z.B. "in der nächsten Konfiguration steht an derselben Bandposition ein a" nicht mehr durch einfaches Schachteln von X-Operatoren ausdrücken, da diese Formel exponentielle Länge hätte und somit die Reduktion selbst nicht mehr in polynomieller Zeit liefe. Obige Aussage lässt sich aber prägnanter ausdrücken, indem man die Position einer Bandzelle mit logarithmisch (in der Länge einer Konfiguration) vielen Propositionen markiert und dann ausdrückt, dass alle späteren Zustände, bei denen zum ersten Mal der Wert aller dieser Propositionen mit den jetzigen Werten übereinstimmt, auch a gilt.

Diese untere Schranke ist auch optimal. Das nächste Resultat präsentieren wir ohne Beweis.

Satz 5.4
Das Erfüllbarkeitsproblem für CTL* ist in 2-EXPTIME.

5.4 Die Logik CTL$^+$

Wegen der hohen Komplexität des Erfüllbarkeitsproblems für CTL* bietet es sich an, nach Fragmenten zu suchen, die eine vernünftige Ausdrucksstärke, aber eine geringere Komplexität haben. CTL ist zwar nicht allzu komplex, dafür kann es aber, wie gezeigt wurde, sehr einfach Aussagen nicht ausdrücken. Der Unterschied zwischen CTL und CTL* ist offensichtlich der, dass in CTL* Pfadoperatoren beliebig tief und beliebig breit geschachtelt werden dürfen, während in CTL unterhalb eines Laufquantors immer genau ein einzige temporaler Operator vorkommt. Lemma 5.3 lässt vermuten, dass die Breite nicht schlimm ist, da sich die Laufquantoren teilweise über boolesche Operatoren hinwegziehen lassen. Wir betrachten somit im folgenden CTL$^+$, ein Fragment von CTL, in dem die temporalen Operatoren nicht beliebig tief geschachtelt werden dürfen – in der Hoffnung, dass wir so eine geringere Komplexität als die von CTL* und eine höhere Ausdrucksstärke als die von CTL erhalten. Dies ist allerdings ein Trugschluss.

Definition 5.5
Die Logik CTL$^+$ besteht aus den Zustandsformeln, die in der folgenden Grammatik aus φ ableitbar sind.

$$\begin{aligned} \varphi &::= q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathtt{A}\psi \mid \mathtt{E}\psi \\ \psi &::= \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \neg\psi \mid \mathtt{X}\varphi \mid \varphi\mathtt{U}\varphi \mid \varphi\mathtt{R}\varphi \end{aligned}$$

Die Semantik ergibt sich wiederum eindeutig als Fragment von CTL*.

Beispiel 5.3
In CTL$^+$ lassen sich die Laufquantoren auf einfache temporale Eigenschaften relativieren. So kann man z.B. sagen, dass jeder Lauf, auf dem immer q gilt, auch irgendwann einmal p erfüllen soll: $\mathtt{A}(\mathtt{G}q \rightarrow \mathtt{F}p)$.

Existentielle Relativierung funktioniert genauso. $\mathtt{E}(q \wedge \mathtt{X}q \wedge (p_1 \mathtt{U} p_2))$ besagt, dass es einen Lauf gibt, der $p_1 \mathtt{U} p_2$ erfüllt, der aber in den ersten beiden Zuständen zusätzlich q erfüllt. Eine Einschränkung aufgrund von "tieferen" temporalen Eigenschaften ist aber nicht möglich. So ist $\mathtt{E}(q \wedge \mathtt{X}q \wedge \mathtt{XX}q \wedge (p_1 \mathtt{U} p_2))$ z.B. keine CTL$^+$-Formel mehr.

5.4.1 Ausdrucksstärke und Prägnanz

Beispiel 5.4
Eine typische CTL$^+$-Eigenschaft quantifiziert über einen Lauf, auf dem mehrere, einfache temporale Eigenschaften gelten. Sei z.B. $\mathcal{P} = \{q_1, \ldots, q_n\}$. Dann lässt sich in CTL$^+$ leicht sagen, dass es einen Lauf gibt, auf dem alle diese Propositionen irgendwann einmal gelten:

$$\varphi \;:=\; \mathtt{E}(\bigwedge_{i=1}^{n} \mathtt{F} q_i)$$

Beachte, dass sich "es gibt einen Lauf, auf dem alle diese Propositionen unendlich oft gelten" nicht in CTL$^+$ ausdrücken lässt, was aus Satz 3.11 und dem folgenden Satz 5.5 folgt. Letzterer verallgemeinert die Beobachtung, dass sich obige Eigenschaft auch in CTL ausdrücken lässt.

Offensichtlich gilt, dass, wenn auf einem Lauf alle Propositionen $q_1, \ldots, q_n$ irgendwann einmal gelten, dann gelten sie auch zeitlich in einer bestimmten Reihenfolge. Dabei vernachlässigen wir die Frage, welche von zweien "zuerst" gilt, wenn beide im selben Zeitpunkt gelten. Dies ist aber genau dann der Fall, wenn es eine Permutation $\sigma : I_n \to I_n$ gibt mit $I_n := \{1, \ldots, n\}$, so dass auf diesem Lauf zuerst $q_{\sigma(1)}$, dann $q_{\sigma(2)}$, usw. gilt. Somit lässt sich obige Eigenschaft in CTL folgendermaßen ausdrücken.

$$\varphi' \;:=\; \bigvee_{\sigma \in S(I_n)} \mathtt{EF}(q_{\sigma(1)} \wedge \mathtt{EF}(q_{\sigma(2)} \wedge \ldots (q_{\sigma(n-1)} \wedge \mathtt{EF} q_{\sigma(n)}) \ldots))$$

wobei S_n die Menge aller Permutationen auf Elementen von I_n darstellt. Beachte, dass $|\varphi| = O(n)$, aber $|\varphi'| = O(n \cdot n!) = 2^{O(n \cdot \log n)}$.

Satz 5.5
CTL $\equiv$ CTL$^+$.

Beweis Die Richtung "$\leq$" gilt trivialerweise bereits aus syntaktischen Gründen. Für die Richtung "$\geq$" sei $\varphi \in$ CTL$^+$. Aufgrund von Lemma 5.2 können wir annehmen, dass alle universellen Laufquantoren in φ durch Negationen und existentielle Quantoren ersetzt wurden. Wir konstruieren nun durch Induktion über $qd(\varphi)$ eine zu φ äquivalente CTL-Formel.

Falls $qd(\varphi) = 0$, dann ist φ lediglich eine boolesche Kombination aus atomaren Propositionen und somit bereits eine CTL-Formel. Sei also $qd(\varphi) > 0$. Wir können o.B.d.A. annehmen, dass φ von der Form $\mathtt{E}\varphi'$ ist, wobei φ' eine boolesche Kombination aus atomaren Propositionen und Formeln der Form $\mathtt{X}\psi$, $\psi \mathtt{U} \psi'$ und $\psi \mathtt{R} \psi'$ ist. Da CTL unter Negation abgeschlossen ist, können wir per Induktionshypothese auch annehmen, dass diese ψ und ψ' bereits CTL-Formeln sind.

Aufgrund der Distributivgesetze für boolesche Operatoren und den Äquivalenzen

- $\psi \mathtt{R} \psi' \equiv \psi' \mathtt{U} (\psi \wedge \psi') \vee \mathtt{G}\psi'$,
- $\mathtt{X}\psi \wedge \mathtt{X}\psi' \equiv \mathtt{X}(\psi \wedge \psi')$,
- $\neg \mathtt{X}\psi \equiv \mathtt{X}\neg\psi$,
- $\mathtt{G}\psi \wedge \mathtt{G}\psi' \equiv \mathtt{G}(\psi \wedge \psi')$,
- $\mathtt{E}(\psi \vee \psi') \equiv \mathtt{E}\psi \vee \mathtt{E}\psi'$
- $\mathtt{E}(\psi \wedge \psi') \equiv \mathtt{E}\psi \wedge \psi'$, falls ψ' ein Literal oder von der Form $\mathtt{E}\ldots$ oder $\neg\mathtt{E}\ldots$ ist,

können wir davon ausgehen, dass φ von der Form

$$\varphi \;=\; \mathtt{E}\big(\mathtt{X}\psi' \wedge \mathtt{G}\psi \wedge \bigwedge_{i=1}^{n} \alpha_i \mathtt{U} \beta_i\big)$$

ist, wobei ψ, ψ' sowie die α_i, β_i bereits CTL- und damit insbesondere Zustandsformeln sind. Dann gilt

$$\varphi \;\equiv\; \bigvee_{I \subseteq \{1,\ldots,n\}} \Big((\bigwedge_{i \in I} \beta_i) \wedge \psi \wedge \mathtt{EX}\Big(\psi' \wedge \underbrace{\mathtt{E}\big(\mathtt{G}\psi \wedge \bigwedge_{i \notin I} \alpha_i \mathtt{U} \beta_i\big)}_{\varphi'} \Big) \Big)$$

Teil φ' ist der einzige, der nicht syntaktisch eine CTL-Formel ist. Ihn kann man aber wie in Bsp. 5.4 äquivalent folgendermaßen in eine CTL-Formel umschreiben.

$$\begin{aligned}
\mathtt{E}(\mathtt{G}\psi \wedge \bigwedge_{i=1}^{n} \alpha_i \mathtt{U} \beta_i) \;\equiv\; \bigvee_{\sigma \in S_n} \mathtt{E}\Big(& (\psi \wedge \bigwedge_{i=1}^{n} \alpha_i)\ \mathtt{U}\ \Big(\beta_{\sigma(1)} \wedge \psi \wedge \\
& \mathtt{E}\Big((\psi \wedge \bigwedge_{i \neq \sigma(1)} \alpha_i)\ \mathtt{U}\ \big(\beta_{\sigma(2)} \wedge \psi \wedge \\
& \quad \ldots \wedge \\
& \quad \mathtt{E}\big((\psi \wedge \alpha_{\sigma(n)})\ \mathtt{U}\ (\beta_{\sigma(n)} \wedge \mathtt{EG}\psi)\big) \ldots \big)\Big)\Big)\Big)
\end{aligned}$$

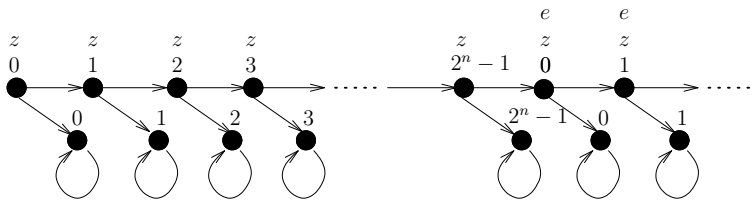
Somit existiert auch eine CTL-Formel, die äquivalent zu φ ist. ∎

Somit ist CTL$^+$ zwar nur so ausdrucksstark wie CTL, aber da die Übersetzung einer CTL$^+$-Formel φ eine CTL-Formel φ' mit $|\varphi'| = 2^{O(n \cdot \log n)}$ erzeugen kann, stellt sich die Frage, ob CTL$^+$ evtl. bestimmte Eigenschaften prägnanter ausdrücken kann als CTL. In anderen Worten: Kann es eine asymptotisch bessere Übersetzung von CTL$^+$ nach CTL geben? Die Antwort ist "nein" [AI01]. Im folgenden zeigen wir durch ein einfaches, modelltheoretisches Argument, dass es mindestens eine subexponentielle (und insbesondere nicht-polynomielle) Schranke an die Größe der Übersetzung gibt.

Satz 5.6
Es gibt CTL$^+$-Formeln φ_n, $n \in \mathbb{N}$, so dass für die kleinsten CTL-Formeln φ'_n, die zu φ_n äquivalent sind, gilt: $|\varphi'_n| = 2^{\Omega(\sqrt{|\varphi_n|})}$.

Beweis Wir konstruieren zuerst CTL$^+$-Formeln φ_n, $n \geq 1$, deren kleinstes Modell doppelt exponentielle Größe hat. Dazu modellieren wir mit Propositionen $c_1, \ldots, c_n$ einen Zähler, der Werte zwischen 0 und $2^n - 1$ annehmen kann. Unter Zuhilfenahme dieses Zählers modellieren wir dann mit einer einzigen Proposition c einen Zähler, der Werte zwischen 0 und $2^{2^n} - 1$ annehmen kann.

Die Idee ist, dass ein Modell von φ_n bisimilar zu dem folgenden Transitionssystem ist.



Dabei kürzen die ganzen Zahlen den Wert der Propositionen $c_1, \ldots, c_n$ in binärer Kodierung ab.

Das Fortschreiten des kleinen Zählers beschreibt die folgende Formel.

$$\varphi_{count} \;:=\; \Big(\bigwedge_{i=1}^{n} \neg c_i\Big) \;\wedge\; \mathtt{AGA}\Big(\; (z \wedge \mathtt{X}z) \rightarrow \bigwedge_{i=1}^{n} \Big((\bigwedge_{j<i} c_j) \vee (c_i \leftrightarrow \mathtt{X}c_i)\Big) \;\wedge\; \Big(\bigvee_{j<i} \neg c_j \vee (c_i \leftrightarrow \mathtt{X}\neg c_i)\Big) \;\Big)$$

Wir benutzen außerdem eine Propositionen z, die nur entlang des langen Laufs gilt, auf dem der Zähler weitergezählt wird. Sie gilt nie mehr, sobald man diesen Lauf verläßt.

$$\varphi_z \;:=\; z \wedge \mathtt{AG}\Big(\big(z \rightarrow (\mathtt{EX}z \wedge \mathtt{EX}\neg z)\big) \;\wedge\; \big(\neg z \rightarrow (\mathtt{AX}\neg z)\big)\Big)$$

Die Beschriftung der Zustände mit den Zählerpropositionen wird eindeutig in die abzweigenden Läufe kopiert. Dadurch kann man später Aussagen über Propositionen auf dem z-Lauf machen, indem man sie alternativ über den abzweigenden Zustand macht.

$$\varphi_{\neg z} \;:=\; \mathtt{AGA}\Big(\mathtt{X}\neg z \rightarrow \big((c \leftrightarrow \mathtt{X}c) \wedge \bigwedge_{i=1}^{n} c_i \leftrightarrow \mathtt{X}c_i\big)\Big)$$

Jetzt benutzen wir jeweils 2^n viele Positionen entlang des z-Laufs, um mit der Proposition c den Wert des großen Zählers festzulegen. Dies ist zu Anfang 0.

$$\varphi_{init} \;:=\; \neg c \wedge \mathtt{A}\Big(\mathtt{G}z \rightarrow \big((\neg c)\mathtt{U}(c \wedge \bigwedge_{i=1}^{n} \neg c_i)\big)\Big)$$

Dieser Zähler wird in der üblichen Weise erhöht beim Übergang zu der nächsten Sequenz von 2^n vielen Zuständen. Solange c gilt bis zur ersten Position, in der c nicht gilt, wird der Wert von c im Zustand 2^n Schritte später vertauscht. In allen Positionen danach wird er dementsprechend erhalten. Wir benutzen eine weitere Proposition k, die markiert, ob ein c-Wert erhalten (k) oder vertauscht ($\neg k$) werden soll.

$$\begin{aligned}\varphi_k \;:=\; \mathtt{AGA}\Big((z \wedge \mathtt{X}z) \;\rightarrow\; \big((\bigwedge_{i=1}^{n} \neg c_i \;\rightarrow\; \neg k) \wedge \\ (\neg k \wedge c \;\rightarrow\; \mathtt{X}\neg k) \wedge \\ (\neg k \wedge \neg c \;\rightarrow\; \mathtt{X}k) \wedge \\ (k \wedge \mathtt{X}(\bigvee_{i=1}^{n} c_i) \;\rightarrow\; \mathtt{X}k)\big)\Big)\end{aligned}$$

Jetzt brauchen wir noch eine Proposition e, die entlang des z-Laufs ihren Wert genau dann ändert, wenn der kleine Zähler den Wert 0 annimmt.

$$\begin{aligned}\varphi_e \;:=\; e \wedge \mathtt{AGA}\Big((z \wedge \mathtt{X}z) \rightarrow \big((\mathtt{X}\bigvee_{i=1}^{n} c_i \;\rightarrow\; (e \leftrightarrow \mathtt{X}e)) \wedge \\ (\mathtt{X}\bigwedge_{i=1}^{n} \neg c_i \;\rightarrow\; (e \leftrightarrow \mathtt{X}\neg e))\big)\Big)\end{aligned}$$

Um das Fortschreiten des c-Zählers zu modellieren, definieren wir zuerst eine Pfadformel, die erfüllt wird von genau den Läufen, die in einem Zustand mit Zählerwert i starten und nach 2^n Schritten im nächsten Zustand mit Zählerwert i in den unteren Teil abbiegen.

$$\begin{aligned}\psi \;:=\; (z \wedge \mathtt{X}z) \;\wedge\; \big(\bigwedge_{i=1}^{n}(c_i \rightarrow \mathtt{F}(\neg z \wedge c_i) \wedge (\neg c_i \rightarrow \mathtt{F}(\neg z \wedge \neg c_i)\big) \;\wedge \\ \Big(\;\Big(\;(\bigvee_{i=1}^{n} c_i) \wedge \neg\big(\mathtt{F}(e \wedge z \wedge \bigwedge_{i=1}^{n} \neg c_i) \wedge \mathtt{F}(\neg e \wedge z \wedge \bigwedge_{i=1}^{n} \neg c_i)\big)\;\Big) \vee \\ \Big(\;(\bigwedge_{i=1}^{n} \neg c_i) \wedge \neg\big(\mathtt{F}(e \wedge z \wedge c_1 \wedge \bigwedge_{i=2}^{n} \neg c_i) \wedge \mathtt{F}(\neg e \wedge z \wedge c_1 \wedge \bigwedge_{i=2}^{n} \neg c_i)\big)\;\Big)\;\Big)\end{aligned}$$

Dann lässt sich das Fortschreiten des c-Zählers so modellieren:

$$\begin{aligned}\varphi_c \;:=\; \mathtt{AGA}\Big(\;\Big(\psi \;\rightarrow\; \Big(k \rightarrow \big((c \rightarrow \mathtt{FAG}c) \wedge (\neg c \rightarrow \mathtt{FAG}\neg c)\big) \wedge \\ \neg k \rightarrow \big((c \rightarrow \mathtt{FAG}\neg c) \wedge (\neg c \rightarrow \mathtt{FAG}c)\big)\Big)\;\Big)\Big)\end{aligned}$$

Dies besagt, dass auf allen relativierten Läufen k zu Beginn gilt, gdw. der c-Wert zu Anfang derselbe ist wie schließlich immer. Betrachtet werden aber nur solche Läufe, die mindestens einen Schritt entlang des z-Lauf machen, nicht zweimal den Wert 0 des

kleinen Zählers entlang dieses Laufs sehen und bei denen der Wert des kleinen Zählers zu Beginn mit dem schließlich angenommenen Wert übereinstimmt. Dies trifft genau auf den Lauf zu, der nach 2^n Schritten vom z-Lauf abbiegt.

Man sieht leicht, dass die Formel

$$\varphi_n \;:=\; \varphi_{count} \wedge \varphi_z \wedge \varphi_{\neg z} \wedge \varphi_{init} \wedge \varphi_k \wedge \varphi_e \wedge \varphi_c$$

erfüllbar ist, aber kein Modell mit weniger als 2^{2^n} vielen Zuständen hat. Laut Satz 5.5 gibt es CTL-Formeln φ'_n, so dass $\varphi'_n \equiv \varphi_n$ für jedes $n \geq 1$. CTL hat jedoch die kleine Modelleigenschaft exponentieller Größe: Wenn φ'_n CTL erfüllbar ist, dann hat es ein Modell der Größe höchstens $|\varphi'_n| \cdot 2^{4 \cdot |\varphi'_n|}$ [EH85]. Da die φ_n jeweils erfüllbar sind, sind auch die φ'_n erfüllbar, haben aber kein Modell der Größe kleiner als 2^{2^n}. Dann gilt:

$$|\varphi'_n| \cdot 2^{4 \cdot |\varphi'_n|} \geq 2^{2^n} \quad \text{gdw.} \quad 5 \cdot |\varphi'_n| \geq 4 \cdot |\varphi'_n| + \log |\varphi'_n| \geq 2^n \quad \text{gdw.} \quad |\varphi'_n| \geq \frac{2^n}{5}$$

Also $|\varphi'_n| = \Omega(2^n)$. Andererseits gilt $|\varphi_n| = O(n^2)$. Ein Vergleich ergibt: $|\varphi'_n| = 2^{\Omega(\sqrt{|\varphi_n|})}$. ■

Wir bemerken, dass sich das Erhöhen eines Zählers auch mit einer Formel der Größe $O(n)$ beschreiben läßt. Somit gilt sogar eine echt exponentielle untere Schranke von $2^{\Omega(n)}$ an die Größe der entsprechenden CTL-Formeln.

5.4.2 Komplexität

Nicht nur, dass die Ausdruckstärke von CTL$^+$ entgegen der Hoffnung gering ist, man kann ebenfalls durch Reduktion auf das Wortproblem für alternierende und exponentiell platzbeschränkte Turing Maschinen zeigen, dass das Erfüllbarkeitsproblem für CTL$^+$ 2-EXPTIME-hart, also genauso schwer wie das für CTL* ist. Dies ist dann natürlich auch eine obere Schranke, da CTL$^+$ ein syntaktisches Fragment von CTL* ist.

Satz 5.7
Das Erfüllbarkeitsproblem für CTL$^+$ ist 2-EXPTIME-vollständig.

Es stellt sich noch die Frage nach dem Model Checking Problem für CTL$^+$. Man sieht leicht, dass dies mindestens so schwer ist wie das Model Checking Problem für LTL$^{\mathsf{min}}$ und somit (vermutlich) schwerer als das Model Checking Problem von CTL.

Satz 5.8
Das Model Checking Problem für CTL$^+$ ist NP-hart und co-NP-hart.

Da CTL$^+$ unter Komplement abgeschlossen ist, d.h. $\mathcal{T}, s \not\models \varphi$ gdw. $\mathcal{T}, s \models \neg\varphi$, können wir nicht erwarten, dass das Model Checking Problem vollständig für NP oder co-NP ist. Man kann zeigen, dass es vollständig für Δ_2^p, eine deterministische Komplexitätsklasse, die oberhalb von NP und co-NP und unterhalb von PSPACE liegt, ist.

5.4.3 Unäres CTL$^+$

Wie bei CTL können wir auch *unäres CTL$^+$*, CTL^{+-}, betrachten. Diese Logik entsteht, wenn man in der Definition von CTL$^+$ die beiden binären, temporalen Operatoren `U` und `R` durch ihre Spezialformen `F` und `G` ersetzt. Offensichtlich gilt CTL$^-$ $\leq$ CTL^{+-} $\leq$ CTL bereits aus syntaktischen Gründen und Satz 5.5. Es stellt sich die Frage, ob diese Inklusionen strikt sind.

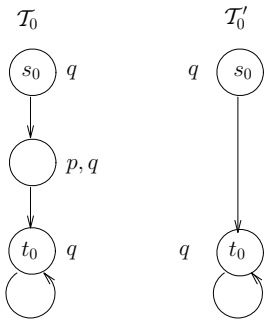
Satz 5.9
CTL^{+-} $\lneq$ CTL.

Beweis In Satz 3.16 wurde gezeigt, dass es keine CTL$^-$-Formel gibt, die äquivalent zu der CTL-Formel $\mathtt{E}(p\mathtt{U}q)$ ist. Der Beweis benutzt zwei Familien von linearen Transitionssystemen, die von keiner CTL$^-$-Formel unterschieden werden können. Auf linearen Transitionssystemen gilt aber sicherlich CTL$^-$ $\equiv$ CTL^{+-}, da sich dort mithilfe von Lemmas 5.3 und 5.4 die Laufquantoren in einer CTL^{+-}-Formel über die booleschen Operatoren ziehen lassen, bis eine CTL$^-$-Formel entstanden ist. Also gibt es auch keine CTL^{+-}-Formel, die die beiden linearen Transitionssysteme unterscheidet. ■

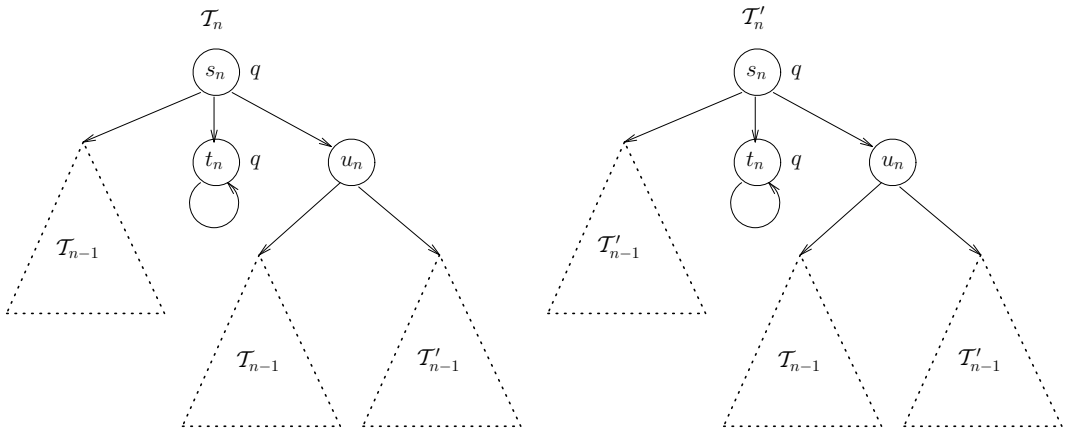
Beachte folgendes Phänomen: Über allgemeinen Transitionssystemen gilt CTL $\equiv$ CTL$^+$, also insbesondere über linearen Modellen. Über solchen gilt auch CTL$^-$ $\equiv$ CTL^{+-}. Es stellt sich die Frage, ob dies nicht auch allgemein so ist. Dem ist aber nicht so.

Satz 5.10
CTL$^-$ $\lneq$ CTL^{+-}.

Beweis Wir zeigen, dass es keine CTL$^-$-Formel gibt, die äquivalent zu der CTL^{+-}-Formel $\mathtt{E}(\mathtt{F}p \wedge \mathtt{G}q)$ ist. Dazu definieren wir wiederum induktiv zwei Familien von Transitionssystemen $\mathcal{T}_n, \mathcal{T}'_n$, $n \in \mathbb{N}$. Für $n = 0$ sehen diese folgendermaßen aus.



Und für $n > 0$ werden sie folgendermaßen konstruiert.



Erstens gilt offensichtlich $\mathcal{T}_n, s_n \models \mathtt{E}(\mathtt{F}p \wedge \mathtt{G}q)$ und $\mathcal{T}'_n, s_n \not\models \mathtt{E}(\mathtt{F}p \wedge \mathtt{G}q)$ für alle $n \in \mathbb{N}$. Ersteres gilt wegen dem Lauf, der rekursiv immer nach $\mathcal{T}_{n-1}$ absteigt und letztendlich $\mathcal{T}_0$ durchläuft. Zweiteres gilt, weil kein Lauf durch ein $\mathcal{T}'_n$, der genauso rekursiv absteigt, immer q erfüllt. Jeder Lauf, der sofort in ein t_n mündet, erfüllt niemals p, und jeder Lauf, der durch ein u_n führt, kann nicht überall q erfüllen.

Wir bemerken, dass außerdem gilt:

1. Alle Zustände t_i in beliebigen $\mathcal{T}_n$ oder $\mathcal{T}'_n$, $n \geq i$, sind bisimilar.
2. $\mathcal{T}_n, u_n \sim \mathcal{T}'_n, u_n$ für alle $n \geq 1$.
3. Jeder Lauf durch $\mathcal{T}_n$ oder $\mathcal{T}'_n$ durchläuft schließlich nur noch einen Zustand t_i für ein $i \leq n$.

Als nächstes zeigen wir durch Induktion über n, dass für alle $\varphi \in \mathrm{CTL}^-$ mit $td(\varphi) \geq n$ gilt: $\mathcal{T}_n, s_n \models \varphi$ gdw. $\mathcal{T}'_n, s_n \models \varphi$. O.B.d.A. können wir davon ausgehen, dass φ nur aus atomaren Propositionen mit Disjunktionen, Negationen und den temporalen Operatoren `EX`, `EG` und `EF` aufgebaut ist. Die Behauptung ist für atomare Propositionen sofort ersichtlich und folgt für die booleschen Operatoren sofort aus der Induktionshypothese. Es bleiben die drei Fälle der temporalen Operatoren übrig.

Fall $\varphi = \mathtt{EX}\psi$. Angenommen es gilt $\mathcal{T}_n, s_n \models \varphi$. Dann gibt es drei Unterfälle: (a) $\mathcal{T}_n, t_n \models \psi$, (b) $\mathcal{T}_n, u_n \models \psi$ oder (c) $\mathcal{T}_{n-1}, s_{n-1} \models \psi$. Liegt (a) oder (b) vor, dann lässt sich sofort mithilfe der Bemerkungen (1) oder (2) schließen, dass auch $\mathcal{T}'_n, t_n \models \psi$ bzw. $\mathcal{T}'_n, u_n \models \psi$ gilt. Liegt Fall (c) vor, dann beachte, dass $td(\psi) = td(\varphi) - 1$ gilt, weswegen sich die Induktionshypothese anwenden lässt und $\mathcal{T}'_{n-1}, s_{n-1} \models \psi$ liefert. Dann gilt aber sicherlich auch $\mathcal{T}'_n, s_n \models \varphi$. Die Rückrichtung läuft vollkommen analog ab.

Fall $\varphi = \mathtt{EG}\psi$. Angenommen es gilt $\mathcal{T}_n, s_n \models \varphi$. Aus der Bemerkung (3) von oben folgt dann insbesondere (a) $\mathcal{T}_n, s_n \models \psi$ und (b) $\mathcal{T}_i, t_i \models \psi$ oder $\mathcal{T}'_i, t_i \models \psi$ für ein $i \leq n$. Jetzt wenden wir die Beobachtung (1) von oben auf (b) an und erhalten auch $\mathcal{T}'_n, t_n \models \psi$. Auf (a) wenden wir die Induktionshypothese an, da $td(\psi) < td(\varphi)$ ist, und erhalten ebenfalls $\mathcal{T}'_n, s_n \models \psi$. Dann gilt aber auch $\mathcal{T}'_n, s_n \models \varphi$. Die Umkehrung wird ebenfalls vollkommen analog bewiesen.

Fall $\varphi = \mathtt{EF}\psi$. Angenommen es gilt $\mathcal{T}_n, s_n \models \varphi$. Dann gibt es also einen Zustand x in $\mathcal{T}_n$, so dass $\mathcal{T}_n, x \models \psi$ gilt, da jeder Zustand in $\mathcal{T}_n$ von s_n aus erreichbar ist. Wir müssen mehrere Fälle unterscheiden.

- Falls $x = s_n$ dann folgt $\mathcal{T}'_n, s_n \models \psi$ aus der Induktionshypothese für ψ.
- Falls $x = u_i$ für ein $i \leq n$, dann folgt $\mathcal{T}'_n, s_n \models \psi$ aus Bemerkung (2) von oben.
- Falls $x = t_i$ für ein $i \leq n$, dann folgt $\mathcal{T}'_n, t_n \models \psi$ aus Bemerkung (1) von oben.
- Falls x von u_n aus erreichbar ist, dann gilt auch $\mathcal{T}'_n, x \models \psi$, da x auch in derselben Form in $\mathcal{T}'_n$ vorhanden ist.
- Falls x von s_{n-1} aus erreichbar ist, dann gibt es ein y, welches von u_n aus erreichbar ist, so dass $x \sim y$. Da der von u_n aus erreichbare Teil ebenfalls in $\mathcal{T}'_n$ vorhanden ist, gilt somit auch hier $\mathcal{T}'_n, y \models \psi$.

In allen Fällen gibt es also einen Zustand y in $\mathcal{T}'_n$, der ψ erfüllt, womit auch $\mathcal{T}'_n, s_n \models \varphi$ gezeigt ist. Die Rückrichtung wird wiederum genauso bewiesen.

Der Rest des Beweises geht wie üblich vor. Angenommen, es gäbe eine CTL$^-$ Formel φ, so dass $\varphi \equiv \mathtt{E}(\mathtt{F}p \wedge \mathtt{G}q)$. Sei $n := td(\varphi)$. Dann müsste $\mathcal{T}_n, s_n \models \varphi$ und $\mathcal{T}'_n, s_n \not\models \varphi$ gelten, was aber der soeben bewiesenen Aussage widerspricht, dass diese beiden nicht von CTL$^-$-Formeln der temporalen Tiefe $\leq n$ unterschieden werden können. ∎

5.5 Fair CTL

Da die Restriktion der Syntax von CTL* auf CTL$^+$ nicht den gewünschten Effekt – insbesondere höhere Ausdrucksstärke als CTL – hatte, erweitern wir jetzt gezielt die Syntax von CTL um wünschenswerte Eigenschaften.

Definition 5.6
Sei $\mathcal{P}$ eine Menge von Propositionen. Ein *Fairnessprädikat* über $\mathcal{P}$ ist eine positive boolesche Kombination Φ von Formeln der Form $\mathtt{GF}l$, wobei l ein Literal über $\mathcal{P}$ ist. Die Syntax von *Fair CTL* (FCTL) lässt in der Syntax von CTL auch um Fairnessprädikate relativierte Laufquantoren zu.

$$\varphi \;:=\; q \mid \neg q \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathtt{EX}\varphi \mid \mathtt{AX}\varphi \mid \mathtt{E}_\Phi(\varphi \mathtt{U} \psi) \mid \mathtt{A}_\Phi(\varphi \mathtt{U} \psi) \mid \mathtt{E}_\Phi(\varphi \mathtt{R} \psi) \mid \mathtt{A}_\Phi(\varphi \mathtt{U} \psi)$$

wobei $q \in \mathcal{P}$ und Φ ein Fairnessprädikat über $\mathcal{P}$ ist. Die temporale Tiefe $td(\varphi)$ einer FCTL-Formel definieren wir hier genauso wie bei CTL – Fairnessprädikate werden also nicht berücksichtigt.

Die Semantik ist eindeutig durch Einbettung in CTL* gegeben:

$$\begin{aligned} \mathtt{E}_\Phi(\psi) &:= \mathtt{E}(\Phi \wedge \psi) \\ \mathtt{A}_\Phi(\psi) &:= \mathtt{A}(\Phi \rightarrow \psi) \end{aligned}$$

Beispiel 5.5
Betrachte ein Szenario, in dem mehrere Prozesse $P_1, \ldots, P_n$ auf eine Ressource zugreifen. Diese darf aber nur von einem einzigen Prozess zur selben Zeit benutzt werden. Jeder Prozess P_i kann über die Proposition s_i signalisieren, dass er auf die Ressource zugreifen möchte. Mit den Propositionen e_i und f_i wird angedeutet, dass Prozess P_i den Zugriff erhält, bzw. die Ressource wieder freigibt. Eine korrekte Implementierung eines solchen Protokolls, welches auch *wechselseitiger Ausschluss* oder *mutual exclusion* genannt wird, sollte sicherlich die folgenden CTL-Formeln erfüllen.

$$\mathtt{AG}\Big(\bigwedge_{i=1}^{n} e_i \rightarrow \mathtt{A}\big(f_i \mathtt{R}(\bigwedge_{j=1}^{n} \neg e_j)\big)\Big)$$

Dies besagt, dass niemals ein Prozess die Ressource erhält, wenn ein anderer sie noch nicht freigegeben hat. Außerdem möchte man sagen, dass jeder Prozess, der die Ressource haben möchte, sie irgendwann auch einmal erhält.

$$\mathtt{AG}\big(\bigwedge_{i=1}^{n} s_i \rightarrow \mathtt{AF}e_i\big)$$

Diese Formel ist aber im Allgemeinen nicht erfüllt, denn es kann evtl. Läufe geben, bei denen ein Prozess die Ressource erhält, sie aber nicht mehr freigibt. Somit kann auf solchen Läufen kein anderer Prozess sie erhalten. Dennoch sollte deswegen die Implementierung des Protokolls, welches lediglich die Signale der Prozesse registriert und die Ressource, wenn möglich, zuteilt, nicht als inkorrekt angesehen werden. Vielmehr ist es der eine Prozess, der die Ressource nicht wieder freigibt, der inkorrekt ist.

FCTL bietet eine verfeinerte Möglichkeit, diese Art der Korrektheit zu spezifizieren. Wir wollen als Fairnessprädikat eine Formel benutzen, die "unendlich oft wird die Ressource einem Prozess zugeteilt" ausdrückt. Beachte, dass gilt:

$$\mathtt{GF}\big(\bigvee_{i=1}^{n} e_i\big) \quad \equiv \quad \bigvee_{i=1}^{n} \mathtt{GF}e_i$$

womit solch ein Fairnessprädikat Φ gefunden ist. Dann lässt sich obige CTL-Formel verfeinern zu

$$\mathtt{AG}\big(\bigwedge_{i=1}^{n} s_i \rightarrow \mathtt{A}_\Phi \mathtt{F}e_i\big)$$

5.5.1 Ausdrucksstärke

Satz 5.11
CTL $\lneq$ FCTL.

Beweis Die Inklusion gilt, da $\mathtt{tt}$ als Fairnessprädikat darstellbar ist, z.B. $\Phi := \mathtt{GF}q \vee \mathtt{GF}\neg q$. Dann ist $\mathtt{E}(\varphi \mathtt{U} \psi) \equiv \mathtt{E}_\Phi(\varphi \mathtt{U} \psi)$, etc.

Die Striktheit der Inklusion ist eine Konsequenz aus Satz 3.11. Angenommen, es würde CTL $\equiv$ FCTL gelten. Dann müsste es auch eine CTL-Formel geben, die äquivalent zu

der FCTL-Formel $\mathtt{E}_{\mathtt{GF}q}(\mathtt{tt}\mathtt{U}\mathtt{tt})$ wäre. Diese ist aber äquivalent zu der CTL*-Formel $\mathtt{EGF}q$, deren Eigenschaft nicht in CTL ausgedrückt werden kann. ■

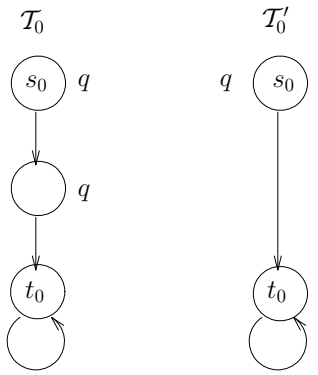
Beachte, dass ein Fairnessprädikat auch eine LTL-Formel ist, weswegen man sie direkt über Läufen interpretieren kann.

Lemma 5.5
Sei $\pi = s_0 s_1 \ldots$ ein Lauf. Für alle Fairnessprädikate Φ und alle $i \in \mathbb{N}$ gilt: $\pi \models \Phi$ gdw. $\pi^{(i)} \models \Phi$.

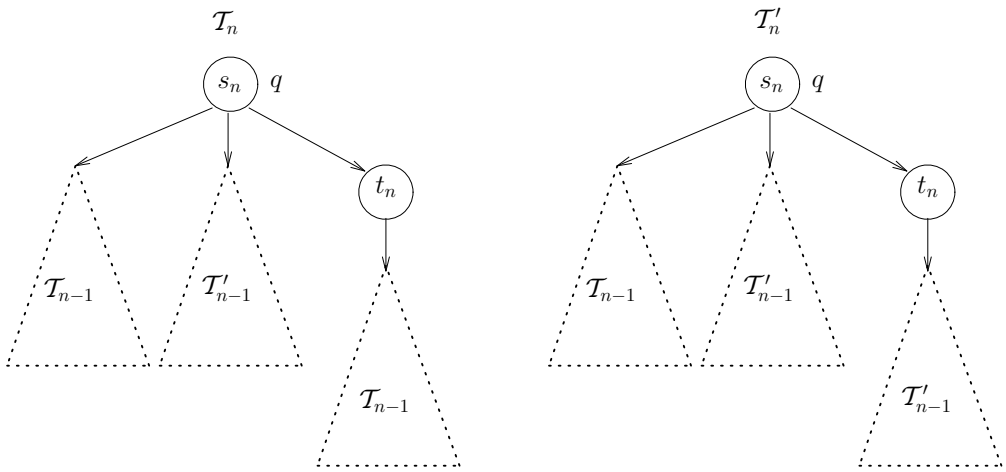
Beweis Dies folgt sofort aus der Tatsache, dass ein endliches Anfangsstück eines Laufs nichts daran ändert, ob auf dem Lauf unendlich oft eine Proposition (nicht) gilt. Es ist also leicht zu sehen, dass für alle Literale l und alle $i \in \mathbb{N}$ gilt: $\pi \models \mathtt{GF}l$ gdw. $\pi^{(i)} \models \mathtt{GF}l$. Für allgemeine Fairnessprädikate folgt die Aussage dann leicht per Induktion über ihren booleschen Aufbau. ■

Satz 5.12
FCTL $\lneq$ CTL*.

Beweis Wir zeigen in der üblichen Weise, dass es keine FCTL-Formel gibt, die äquivalent zu der CTL*-Formel $\mathtt{AF}(q \wedge \mathtt{X}q)$ ist. Dazu konstruieren wir wiederum zwei Familien von Transitionssystemen $\mathcal{T}_n$ und $\mathcal{T}'_n$ für alle $n \in \mathbb{N}$ wie folgt. Für $n = 0$ sehen diese folgendermaßen aus.



Für $n > 0$ sind diese induktiv definiert als



Man erkennt leicht, dass für alle $n \in \mathbb{N}$ gilt: $\mathcal{T}_n, s_n \models \mathtt{AF}(q \wedge \mathtt{X}q)$, aber $\mathcal{T}'_n, s_n \not\models \mathtt{AF}(q \wedge \mathtt{X}q)$.

Als nächstes zeigen wir durch Induktion über den Aufbau von FCTL-Formeln φ, dass für alle $n \in \mathbb{N}$ mit $td(\varphi) \leq n$ gilt: $\mathcal{T}_n, x \models \varphi$ gdw. $\mathcal{T}'_n, x \models \varphi$, wobei $x \in \{s_n, t_n\}$. Die Aussage des Satzes ergibt sich daraus dann wieder in der üblichen Weise.

Für atomare Propositionen ist dies wiederum sofort ersichtlich, und die Fälle der booleschen Operatoren werden sofort aus der Induktionshypothese hergeleitet. Es bleiben noch die Fälle $\mathtt{EX}\psi$, $\mathtt{E}_\Phi(\psi_1 \mathtt{U} \psi_2)$ und $\mathtt{E}_\Phi(\psi_1 \mathtt{R} \psi_2)$ für beliebige Fairnessprädikate Φ zu zeigen. Wir beschränken uns hier auf die Zustände s_n. Für die t_n wird dies alles analog bewiesen.

Fall $\varphi = \mathtt{EX}\psi$: Es gilt $\mathcal{T}_n, s_n \models \varphi$ gdw. $\mathcal{T}_n, t_n \models \psi$ oder $\mathcal{T}_{n-1}, s_{n-1} \models \psi$ oder $\mathcal{T}'_{n-1}, s_{n-1} \models \psi$. Auf den ersten Fall lässt sich die Induktionshypothese anwenden. Damit erhält man sofort $\mathcal{T}'_n, s_n \models \varphi$ und umgekehrt.

Für die verbleibenden Fälle bemerken wir folgendes. Sei Φ ein beliebiges Fairnessprädikat. Dann gilt für alle $n \in \mathbb{N}$ und für alle Läufe π, π' in $\mathcal{T}_n$ oder $\mathcal{T}'_n$: $\pi \models \Phi$ gdw. $\pi' \models \Phi$. Dies ist eine Konsequenz aus Lemma 5.5, da alle solche Läufe von der Form $\sigma(t_0)^\omega$ für ein beliebiges Präfix σ sind, und sich somit zwei verschiedene Läufe nur durch ein endliches Anfangsstück unterscheiden.

Sei Φ also ein beliebiges Fairnessprädikat. Dann sind entweder alle Läufe in diesen Transitionssystemen oder keiner Modell von Φ. D.h. über diesen Familien von Transitionssystemen ist FCTL nur so ausdrucksstark wie CTL und es reicht aus, die übrigen Fälle durch die einfachen CTL-Konstrukte $\mathtt{E}(\psi_1 \mathtt{U} \psi_2)$ bzw. $\mathtt{E}(\psi_1 \mathtt{R} \psi_2)$ zu ersetzen.

Fall $\varphi = \mathtt{E}(\psi_1 \mathtt{U} \psi_2)$: Angenommen $\mathcal{T}_n, s_n \models \mathtt{E}(\psi_1 \mathtt{U} \psi_2)$. Dann gilt $\mathcal{T}_n, s_n \models \psi_2$ oder es gibt einen Lauf π, der $\psi_1 \mathtt{U} \psi_2$ erfüllt. Im ersten Fall erhatlen wir $\mathcal{T}'_n, s_n \models \varphi$ direkt aus der Induktionshypothese. Ansonsten müssen wir drei Fälle unterscheiden. Falls π sofort in $\mathcal{T}_{n-1}$ oder $\mathcal{T}'_{n-1}$ mündet, dann finden wir denselben Lauf auch in $\mathcal{T}'_n$ und können daraus ebenfalls $\mathcal{T}'_n, s_n \models \varphi$ schließen. Falls π über t_n in $\mathcal{T}_{n-1}$ mündet, dann gibt es wiederum zwei Unterfälle: Wenn $t_n \models \psi_2$, dann gilt mit der Induktionshypothese für t_n und ψ_2

sowie für s_n und ψ_1 auch $\mathcal{T}'_n \models \varphi$. Wenn ψ_2 erst in $\mathcal{T}_{n-1}$ erfüllt wird, dann findet sich aber auch ein Pfad, der sofort – und nicht über t_n – in $\mathcal{T}_{n-1}$ mündet und $\psi_1 \mathtt{U} \psi_2$ erfüllt. Dieser existiert aber auch in $\mathcal{T}'_n$, womit die Behauptung auch in diesem Unterfall bewiesen ist. Die Rückrichtung wird analog gezeigt.

Fall $\varphi = \mathtt{E}(\psi_1 \mathtt{R} \psi_2)$: Dies geht genauso durch multiple Fallunterscheidungen wie im vorherigen Fall. ■

5.5.2 Komplexität

Es zeigt sich, dass FCTL eine bessere Einschränkung von CTL* bzgl. Komplexität und Ausdrucksstärke als CTL$^+$ ist. Aus den Sätzen 5.5 und 5.11 folgt natürlich CTL$^+ \leq$ FCTL. Andererseits kann man zeigen, dass die Model Checking Komplexität von FCTL gleich der von CTL$^+$, nämlich Δ_2-vollständig ist. Insbesondere gilt das folgende Resultat.

Satz 5.13
Model Checking FCTL ist NP-hart und co-NP-hart.

Dass dies nicht an den Fairnessoperatoren $\mathtt{GF}l$ alleine liegt, zeigt folgendes Resultat.

Satz 5.14
Das Model Checking Problem für FCTL mit Fairnessprädikaten der Form $\bigwedge_i \bigvee_j \mathtt{GF} l_{ij}$ ist P-vollständig.

Dass man durch diese eingeschränkte Form keine Ausdrucksstärke verliert, sollte klar sein: Jedes Fairnessprädikat lässt sich äquivalent in diese Form übersetzen. Dabei kann allerdings die resultierende Formel exponentiell größer werden.