

2.1 Lineare Temporallogiken: LTL

$\mathbb{N}$ bezeichne die Menge der nicht-negativen ganzen Zahlen (inklusive der Null).

Sei Σ ein Alphabet. Ein *endliches Wort* über Σ ist eine endliche Folge $a_1 a_2 \dots a_n$, sodass $a_i \in \Sigma$ für alle $i \in \{1, \dots, n\}$. Wir schreiben Σ^* für die Menge aller endlichen Wörter über Σ .

Ein *unendliches Wort* (auch: ω -Wort) über Σ ist eine Funktion $w : \mathbb{N} \setminus \{0\} \rightarrow \Sigma$. Alternativ betrachten wir ein unendliches Wort als eine unendliche Folge $a_1 a_2 \dots$, sodass $a_i \in \Sigma$ für alle $i \geq 1$. Wir schreiben Σ^ω für die Menge aller unendlichen Wörter.

Wir definieren weiterhin $\Sigma^\infty := \Sigma^* \cup \Sigma^\omega$. Die Länge eines Wortes w bezeichnen wir mit $|w|$, wobei wir $|w| := \infty$ setzen, falls w unendlich ist.

Sei nun AP eine abzählbare Menge atomarer propositionaler Variablen. Die Menge der LTL-Formeln über AP ist durch folgende Grammatik definiert:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid \text{X}\varphi \mid \varphi \text{U}\varphi$$

Hierbei sind $\neg$ und $\vee$ die aus der klassischen Aussagenlogik bekannten Booleschen Operatoren *Negation* und *Disjunktion*, und X ist ein einstelliger temporaler Operator *Next*, und U ist ein zweistelliger temporaler Operator *Until*.

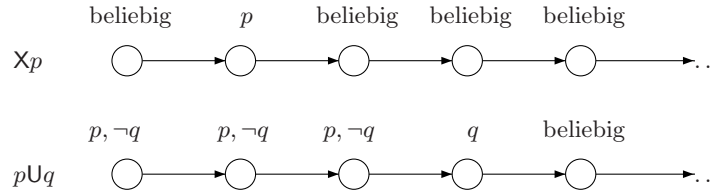
LTL-Formeln werden über Wörtern (endliche und unendliche) über dem Alphabet $\Sigma = 2^{\text{AP}}$ interpretiert (Achtung: ein Buchstabe entspricht einer Menge von propositionalen Variablen in AP .) Sei $w = P_1 P_2 \dots \in \Sigma^\infty$. Weiterhin sei $i \in \{1 \dots |w|\}$ eine Position im Wort w . Wir definieren im Folgenden die Relation $\models$, die beschreibt wann w und i eine LTL-Formel erfüllen:

- $(w, i) \models p$ gdw. $p \in P_i$
- $(w, i) \models \neg \varphi$ gdw. $(w, i) \not\models \varphi$
- $(w, i) \models \varphi_1 \vee \varphi_2$ gdw. $(w, i) \models \varphi_1$ oder $(w, i) \models \varphi_2$
- $(w, i) \models \text{X}\varphi$ gdw. $i + 1 \leq |w|$, $(w, i + 1) \models \varphi$
- $(w, i) \models \varphi_1 \text{U}\varphi_2$ gdw. es existiert eine Position j mit $i \leq j \leq |w|$ sodass $(w, j) \models \varphi_2$ und $(w, k) \models \varphi_1$ für alle $i \leq k < j$.

Wir sagen dass ein Wort w eine LTL-Formel *erfüllt* (geschrieben: $w \models \varphi$), falls $(w, 1) \models \varphi$. In diesem Fall nennen wir w ein *Modell von φ* . Wir unterscheiden gegebenenfalls endliche und unendliche Modelle, z.B. auch bei der Definition der semantischen Äquivalenz im Folgenden: Zwei LTL-Formeln φ und ψ sind *semantisch äquivalent für endliche Wörter*, geschrieben $\varphi \equiv^{fin} \psi$, falls für jedes endliche Wort $w \in \Sigma^*$ gilt: $w \models \varphi$ gdw. $w \models \psi$. Analog definieren wir den Begriff der semantischen Äquivalenz für unendliche Wörter und schreiben $\varphi \equiv^\omega \psi$. Falls sowohl $\varphi \equiv^{fin} \psi$ als auch $\varphi \equiv^\omega \psi$ gilt, so schreiben wir $\varphi \equiv \psi$.

Im Folgenden skizzieren wir die intuitive Bedeutung der temporalen Operatoren X und U für den einfachen Fall, dass $\text{AP} = \{p, q\}$.

Die Formel $\text{X}p$ erfordert, dass in der nächsten Position p gilt. Die Formel $p\text{U}q$ erfordert, dass beginnend in der aktuellen Position, an jeder Position p gilt, bis schliesslich q gilt.



Einstellige Operatoren binden stärker als zweistellige Operatoren. $\neg$ und X binden gleich stark. U bindet stärker als $\vee$. Klammern können dementsprechend wann immer möglich weggelassen werden.

Wir verwenden folgende Operatoren als Abkürzungen:

- $\varphi_1 \wedge \varphi_2 := \neg(\neg\varphi_1 \vee \neg\varphi_2)$,
- $\varphi_1 \rightarrow \varphi_2 := \neg\varphi_1 \vee \varphi_2$
- **true** $:= p \vee \neg p$,
- **false** $:= \neg\text{true}$,
- $F\varphi := \text{true}U\varphi$,
- $G\varphi := \neg F\neg\varphi$.

Die ersten vier Abkürzungen sind Ihnen aus der Klassischen Aussagenlogik bekannt. Der einstellige temporale Operator F steht für *Finally*, und der einstellige temporale Operator G steht für *Globally*. In der Literatur wird für F oft auch $\Diamond$, und für G das Zeichen $\Box$ verwendet (siehe Modallogik).

1. Übung. Geben Sie für $w = P_1P_2\dots$ und $i \in \{1, \dots, |w|\}$ an, wann $(w, i) \models \varphi$ gilt für die Fälle (a) $\varphi = F\varphi'$ und (b) $\varphi = G\varphi'$. Zeichnen Sie dann in einer Skizze wie oben die intuitive Bedeutung der beiden Operatoren ein.

Beispiele

- Die safety property, die aussagt, dass zwei Prozesse P_0 und P_1 sich nicht gleichzeitig in ihren jeweiligen kritischen Bereichen aufhalten sollten, kann (bei entsprechender Wahl von AP) durch die LTL-Formel

$$G(\neg\text{crit}_0 \vee \neg\text{crit}_1)$$

ausgedrückt werden.

- Die liveness property, die aussagt, dass jeder wartende Prozess irgendwann einmal in seinem kritischen Bereich ist (starvation freedom), kann so ausgedrückt werden:

$$(G\text{Fwait}_0 \rightarrow G\text{Fcrit}_0) \wedge (G\text{Fwait}_1 \rightarrow G\text{Fcrit}_1)$$

2. Übung. Geben Sie für die etwas stärkere liveness property, die aussagt, dass jeder Prozess sich unendlich oft in seinem kritischen Bereich aufhält, eine LTL-Formel an.

3. Übung. Handelt es sich bei den folgenden Formeln um eine safety property oder um eine liveness property?

1. $G(\text{alarm} \rightarrow F\text{stop})$
2. $G\neg\text{alarm}$

4. Übung. Welche der folgenden LTL-Formeln sind miteinander semantisch äquivalent? Beweisen Sie Ihre Antwort!

- $GF\phi \rightarrow GF\psi$,
- $G(F\phi \rightarrow F\psi)$,
- $GF(\phi \rightarrow \psi)$,

5. Übung. Zeigen Sie, dass die beiden folgenden LTL-Formeln nicht semantisch äquivalent sind.

- $G\varphi_1 \vee (\varphi_1 U \varphi_2)$
- $\neg(\neg\varphi_2 U \neg\varphi_1)$

Unsere Definition der Semantik geht von einem *reflexiven Until* aus, in dem der aktuelle Zeitpunkt als Teil der Zukunft angenommen wird. In der Programmierung ist dies eine gängige Annahme, jedoch gibt es auch Arbeiten, in denen *Until* in einem strengen, irreflexiven Sinne verwendet wird, im Folgenden als U_{strict} bezeichnet:

- $(w, i) \models \varphi_1 U_{\text{strict}} \varphi_2$ gdw. es existiert eine Position j mit $i < j \leq |w|$ sodass $(w, j) \models \varphi_2$ und $(w, k) \models \varphi_1$ for all $i < k < j$.

6. Übung. Zeigen Sie, dass Sie sowohl X als auch U durch U_{strict} darstellen können, d.h. dass es sowohl für Formeln der Form $X\varphi$ als auch für Formeln der Form $\varphi_1 U \varphi_2$ eine semantisch äquivalente Formel gibt, die als temporalen Operator lediglich den strikten Untiloperator verwendet.

Falls Sie Übung 6 gelöst haben, so haben Sie gezeigt, dass LTL_{strict} definiert durch die Grammatik

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi U_{\text{strict}} \varphi$$

mindestens so ausdrucksstark wie LTL (wie oben definiert) ist; dazu gegebenenfalls später in der Vorlesung mehr.

7. Übung. Wir definieren den temporalen zweistelligen Operator R wie folgt:

$$\varphi_1 R \varphi_2 := \neg(\neg\varphi_1 U \neg\varphi_2).$$

Geben Sie die Semantik dieser Operatoren explizit an. Wie würden Sie eine Formel der Form $\varphi_1 R \varphi_2$ in die natürliche Sprache übersetzen? Zeichnen Sie auch wieder in einer Skizze wie oben die intuitive Bedeutung dieser neuen Operatoren ein.

Eine LTL-Formel ist in *Negationsnormalform* (NNF), falls alle in der Formel vorkommenden Negationszeichen nur vor propositionalen Variablen auftreten. Jede LTL-Formel kann durch die Anwendung folgender semantischer Äquivalenzen in eine semantisch äquivalente LTL-Formel in NNF überführt werden. Man beachte jedoch, dass hierzu die Operatoren $\wedge$ und R notwendig sind.

- $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$
- $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$
- $\neg(\varphi U \psi) \equiv \neg\varphi R \neg\psi$
- $\neg(\varphi R \psi) \equiv \neg\varphi U \neg\psi$
- $\neg X\psi \equiv X\neg\psi$
- $\neg\neg\varphi \equiv \varphi$

Man beachte weiterhin, dass sich bei der Überführung in NNF die Anzahl der in den Formeln vorkommenden temporalen Operatoren nicht verändert.

Die folgenden semantischen Äquivalenzen reduzieren (von links nach rechts angewendet) die Anzahl der temporalen Operatoren. Dies wird später von Nutzen sein.

- $(X\varphi) \wedge (X\psi) \equiv X(\varphi \wedge \psi)$
- $(X\varphi) U (X\psi) \equiv X(\varphi U \psi)$
- $(\varphi R \psi_1) \wedge (\varphi R \psi_2) \equiv \varphi R (\psi_1 \wedge \psi_2)$
- $(\varphi_1 R \psi) \vee (\varphi_2 R \psi) \equiv (\varphi_1 \vee \varphi_2) R \psi$
- $(G\varphi) \wedge (G\psi) \equiv G(\varphi \wedge \psi)$
- $GF\varphi \vee GF\psi \equiv GF(\varphi \vee \psi)$.

8. Übung. In der obigen Auflistung fehlen einige offensichtliche Fälle. Geben Sie mindestens drei solcher Fälle an und prüfen Sie, ob es sich dabei wirklich um semantische Äquivalenzen handelt.

9. Übung. Können die folgenden natürlichsprachlichen Konstrukte durch LTL-Formeln ausgedrückt werden?

1. Prozess 0 ist bald wieder in seinem kritischen Bereich.
2. Prozess 0 ist solange in seinem kritischen Bereich bis Prozess 1 in seinem kritischen Bereich ist, und falls Prozess 1 gar nicht mehr in seinen kritischen Bereich eintritt, ist Prozess 0 für immer in seinem kritischen Bereich.
3. Prozess 0 ist fast immer in seinem kritischen Bereich.
4. Prozess 0 ist die kommenden drei Zeitpunkte in seinem kritischen Bereich.
5. Prozess 0 ist spätestens in drei Sekunden in seinem kritischen Bereich.
6. Prozess 0 ist seit drei Zeitpunkten in seinem kritischen Bereich.

7. Entweder, Prozess 0 ist irgendwann in seinem kritischen Bereich und bis dahin ist Prozess 1 in seinem kritischen Bereich, oder Prozess 1 ist für immer in seinem kritischen Bereich.

Die *Grösse* einer LTL-Formel φ sei definiert durch die Anzahl der in φ vorkommenden Operatoren, und sei durch $|\varphi|$ denotiert.

10. Übung. Definieren Sie $|\varphi|$ induktiv über Formelaufbau.

Im Folgenden definieren wir die durch eine LTL-Formel definierte Sprache. Wir unterscheiden hier wieder endliche und unendliche Modelle: Sei φ eine LTL-Formel. Definiere $L^{fin}(\varphi) := \{w \in \Sigma^* \mid w \models \varphi\}$, und $L^\omega(\varphi) := \{w \in \Sigma^\omega \mid w \models \varphi\}$. Wir definieren das *Erfüllbarkeitsproblem für LTL über endlichen Wörtern* wie folgt.

Erfüllbarkeitsproblem für LTL

EINGABE: LTL-Formel φ

FRAGE: Gilt $L(\varphi) \neq \emptyset$, d.h., gibt es ein Wort $w \in \Sigma^*$ sodass $w \models \varphi$?

Analog definieren wir das Erfüllbarkeitsproblem für LTL über ω -Wörtern.

Letzte Übung. Geben Sie je eine LTL-Formel φ an, für die (a) kein endliches Modell aber ein unendliches Modell bzw. (b) kein unendliches Modell aber ein endliches Modell existiert.

LTL wird heute erfolgreich als Spezifikationssprache in Softwarewerkzeugen wie SPIN und SMV verwendet.

Wir definieren abschliessend für LTL-Formeln φ :

- $\text{Teil}(\varphi)$ bezeichne die Menge aller Teilformeln von φ .
- $\text{X}(\varphi)$ bezeichne die Menge aller Teilformeln der Form $\text{X}\psi$.
- $\text{U}(\varphi)$ bezeichne die Menge aller Teilformeln der Form $\psi_1 \text{U} \psi_2$.
- $\text{AP}(\varphi)$ bezeichne die Menge der in φ vorkommenden propositionalen Variablen.
- $\text{Lit}(\varphi)$ bezeichne die Menge der in φ vorkommenden Literale, d.h. Teilformeln der Form p oder $\neg p$ mit $p \in \text{AP}$.