Temporální logika

Temporální logika poskytuje rámec a prostředky pro analýzu dynamických (imperativních) stavových systémů a hraje zde stejnou úlohu jakou má klasická logika pro matematické systé-my.

V intuitivním smyslu stavové systémy zahrnují „stavy“ a vyka-zují „chování“ při průchodu posloupnostmi takových stavů.

Máme na mysli například, programové moduly, komunikační protokoly, databázové systémy, logické obvody, čipy a obecně výpočetní procesy, které při provádění procházejí určitými stavy a vykazují specifické chování.

Podobně jako v klasické logice je vhodné začít s výrokovu versí temporální logiky.

Ingredience

Stavy. K popisu stavů použijeme výrokové konstanty místo výrokových proměnných. Výrokové konstanty vyjadřují jednoduchá tvrzeni například „globální proměnná x má hodnotu 10“.

K popisu všech stavů použijeme množinu výrokových konstant, která může být konečná i nekonečná. Stav je určen pravdivostním ohodnocením všech výrokových konstant.

K popisu konkrétních systémů obvykle postačí konečná množina výrokových konstant.

Čas. Podobně jako výpočetní procesy postupují v jednotlivých krocích, uvažujeme čas diskretní sestávající z jednotlivých časových bodů od počátečního bodu (okamžiku) k dalším (budoucím) časovým bodům.

Budoucnost systému lze modelovat různým uspořádáním časových bodů. Nejjednodušším a vcelku realistickým předpokladem je uspořádání časových bodů do lineární množiny.

V takovém případě budou časové body tvořit konečnou nebo nekonečnou posloupnost, kterou očíslujeme přirozenými čísly

$$m_0 , m_1 , m_2 , m_3 , \ \ldots , m_n , m_{n+1} , \ \ldots$$

Takové uspořádání používá Výroková lineární temporální logika (LTL).

Při popisu složitějších systémů popisujících například distribuované výpočty se používají i jiná uspořádání časových bodů například ve tvaru stromu. Pak mluvíme o větvícím se čase.

Je-li V množina výrokových konstant, jazyk výrokové temporální logiky L_{LTL} sestává ze

- všech výrokových konstant z množiny V a
- symbolů **false** **->** □ ○ ( )

Definice. (Formule LTL)

1. Každá výroková konstanta z množiny V je formule.
2. **false** je formule.
3. Jsou-li A a B formule, potom $A \rightarrow B$ je formule.
4. Je-li A formule, potom $\circ A$ a $\Box A$ jsou formule.

Ostatní spojky se zavádí jako zkratky

$\neg A$ je zkratka za formuli **false** -> A

true je zkratka za formuli $\neg$ **false**

$\vee$, $\wedge$, <-> jako v klasické logice ,

$\Diamond A$ je zkratka za formuli $\neg \Box \neg A$

Temporální operátory $\circ$, $\Box$, $\Diamond$ se (anglicky) nazývají *nexttime* nebo jen *next* , *always* nebo *henceforth* a *sometime.*

Formule $\circ A$, $\Box A$ a $\Diamond A$ se (anglicky) čtou *nextA* , *alwaysA* a *sometimeA* . Česky *příště A* , *vždy A* a *někdy A* .

Preference $\neg$, $\circ$, $\Box$, $\Diamond$ váží silněji než $\vee$, $\wedge$, -> a <-> má nejslabší prioritu.

V klasické logice se pravdivostní hodnoty výrokových formulí určují z pravdivostního ohodnocení výrokových proměnných (tedy v jedné interpretaci, v jednom ‘stavu’) .

V temporální logice, kde se výrokové formule tvoří z výrokových konstant se pravdivostní hodnoty určují z více pravdivostních ohodnocení těchto konstant (tedy ve více interpretacích , ve více ‘stavech’) .

Pro ‘stav’ používáme jako synonymum ‘svět’ a sémantika temporální logiky je definována pomocí Kripkeovy sémantiky ‘možných světů’ .

Sémantika pro Lineární Temporální Logiku LTL

Nechť V je množina výrokových konstant.

Temporální (nebo Kripkeova) *struktura* pro V je nekonečná posloupnost $\mathbf{K} = (\eta_0 , \eta_1 , \eta_2 , \ldots)$ pravdivostních ohodnocení

$$\eta_i : V \rightarrow \{\mathrm{ff} , \mathrm{tt} \}$$

které nazýváme *stavy*.

Říkáme, že η_0 je *počáteční stav* $\mathbf{K}$ v časovém bodu m_0 a že η_{n+1} je následující stav ke stavu η_n .

Stavy jsou tedy pravdivostní ohodnocení množiny výrokových konstant V a popisují ‘stav světa ‘ v časových okamžicích (časových bodech)

$$m_0, m_1, m_2, m_3, \ldots, m_n, m_{n+1}, \ldots$$

Pro temporální strukturu $\mathbf{K}$, index i a formuli A *induktivně* definujeme pravdivostní hodnotu $\mathbf{K}_i(A)$, neformálně, pravdivostní hodnotu formule A v časovém bodě m_i.

$\mathbf{K}_i(v) = \eta_i(v)$ $\qquad v \in V$

$\mathbf{K}_i(\mathbf{false}) = \mathrm{ff}$

$\mathbf{K}_i(A \text{ -> } B) = \mathrm{tt}$ *právě když*

$\mathbf{K}_i(A) = \mathrm{ff}$ *nebo* $\mathbf{K}_i(B) = \mathrm{tt}$

Pro operátory

$\mathbf{K}_i(\circ A) = \mathbf{K}_{i+1}(A)$

$\mathbf{K}_i(\Box A) =$ tt *právě když* $\mathbf{K}_j(A) =$ tt *pro každé* $j \geq i$

Pro definované symboly

$\mathbf{K}_i(\neg A) =$ tt *právě když* $\mathbf{K}_i(A) =$ ff

$\mathbf{K}_i(A \vee B) =$ tt *právě když*

$\mathbf{K}_i(A) =$ tt *nebo* $\mathbf{K}_i(B) =$ tt

.

$\mathbf{K}_i(A \wedge B) =$ tt *právě když*

$\mathbf{K}_i(A) =$ tt *a* $\mathbf{K}_i(B) =$ tt

.

$\mathbf{K}_i(A <-> B) = \text{tt}$ *právě když* $\mathbf{K}_i(A) = \mathbf{K}_i(B)$

$\mathbf{K}_i(\mathbf{true}) = \text{tt}$

$\mathbf{K}_i(\Diamond A) = \text{tt}$ *právě když* $\mathbf{K}_j(A) = \text{tt}$ *pro nějaké* $j \geq i$

Povšimněme si , že pravdivostní hodnoty $\mathbf{K}_i(\Box A)$ a $\mathbf{K}_i(\circ A)$ jsou definovány pomocí následujících stavů a aktuálního stavu.

Platí

$\mathbf{K}_i(\Diamond A) = \text{tt}$ *právě když* $\mathbf{K}_i(\neg \Box \neg A) = \text{tt}$

právě když $\mathbf{K}_i(\Box \neg A) = \text{ff}$

právě když $\mathbf{K}_j(\neg A) = \text{ff}$ *pro nějaké* $j \geq i$

právě když $\mathbf{K}_j(A) = \text{tt}$ *pro nějaké* $j \geq i$

Definice (validita, sémantický důsledek)

Nechť A je formule jazyka logiky $LTL(V)$ a T je množina formulí stejného jazyka.

Říkáme, že A je *validní* v temporální struktuře $\mathbf{K}$ pro V, (nebo že A je splněna v $\mathbf{K}$) a píšeme $\mathbf{K} \models A$ nebo $\models_{\mathbf{K}} A$, jestliže $\mathbf{K}_i(A) = \mathrm{tt}$ *pro všechna* i

Říkáme, že struktura $\mathbf{K}$ je modelem množiny formulí T, jestliže $\mathbf{K} \models B$ pro všechny formule B z T.

Říkáme, že A je (sémantický) *důsledek* T a píšeme $T \models A$, jestliže A je validní v každém modelu $\mathbf{K}$ množiny T.

Říkáme, že A je validní a píšeme $\models A$, jestliže A je validní v každé temporální struktuře $\mathbf{K}$.
Jinými slovy, A je validní, jestliže $\emptyset \models A$.

Příklad. $\neg\circ A \leftrightarrow \circ\neg A$ je validní formule.

Je třeba ukázat, že $\mathbf{K}_i(\neg\circ A) = \mathbf{K}_i(\circ\neg A)$ platí pro každou strukturu $\mathbf{K}$ a všechny časové body i.

$$\mathbf{K}_i(\neg\circ A) = \mathrm{tt} \Leftrightarrow \mathbf{K}_i(\circ A) = \mathrm{ff}$$
$$\Leftrightarrow \mathbf{K}_{i+1}(A) = \mathrm{ff}$$
$$\Leftrightarrow \mathbf{K}_{i+1}(\neg A) = \mathrm{tt}$$
$$\Leftrightarrow \mathbf{K}_i(\circ\neg A) = \mathrm{tt}$$

Lemma 1. (korektnost pravidla modus ponens)

Nechť $\mathbf{K}$ je temporální struktura a $i \in N$, nechť
$\mathbf{K}_i(A) = \text{tt}$ a $\mathbf{K}_i(A \rightarrow B) = \text{tt}$, potom $\mathbf{K}_i(B) = \text{tt}$.

Důkaz. $\mathbf{K}_i(A \rightarrow B) = \text{tt}$, tedy $\mathbf{K}_i(A) = \text{ff}$ nebo $\mathbf{K}_i(B) = \text{tt}$.

Z předpokladu $\mathbf{K}_i(A) = \text{tt}$ dostáváme $\mathbf{K}_i(B) = \text{tt}$.

Věta 2. Je-li $T \models A$ a $T \models A \rightarrow B$ potom $T \models B$.

Důkaz. Nechť struktura $\mathbf{K}$ je modelem T. Potom pro každé i platí

$$\mathbf{K}_i(A) = \mathbf{K}_i(A \rightarrow B) = \text{tt}$$

a podle lemmatu pak také $\mathbf{K}_i(B) = \text{tt}$. Tedy $T \models B$.

Věta 3. Je-li $T \models A$ potom $T \models \circ A$ a $T \models \Box A$.
Speciálně $A \models \circ A$ a $A \models \Box A$.

Důkaz. Nechť **K** je libovolná temporální struktura, která je modelem T a nechť i je přirozené číslo.

Podle předpokladu platí $\mathrm{K}_j(A)$ pro všechna j, tedy také

$$\mathrm{K}_{i+1}(A) = \mathrm{tt} \quad \text{a} \quad \mathrm{K}_j(A) = \mathrm{tt} \text{ pro všechna } j,\ j \geq i .$$

To znamená, že

$$T \models \circ A \qquad \text{a} \qquad T \models \Box A .$$

Zajímavější je následující tvrzení, které uvádíme bez důkazu.

Věta 4. Je-li $T \models A \rightarrow B$ a $T \models A \rightarrow \circ A$,

potom $T \models A \rightarrow \Box B$.

Označení. Nechť $\mathbf{K} = (\eta_0 , \eta_1 , \eta_2 , \ldots)$ je nějaká temporální struktura pro množinu výrokových konstant V. Nechť i je přirozené číslo. Temporální strukturu $\mathbf{K}^i$, která vznikne z $\mathbf{K}$ posunutím času o i kroků do budoucnosti, definujeme takto:

$$\mathbf{K}^i = (\eta_0' , \eta_1', \eta_2', \ldots)$$

kde $\eta_j' = \eta_{i+j}$ pro každé j. $\mathbf{K}^i$ je také temporální struktura podle původní definice, ale budeme ji úsporněji zapisovat jako

$$\mathbf{K}^i = (\eta_i , \eta_{i+1} , \eta_{i+2} , \ldots , \eta_{i+j} , \eta_{i+j+1} , \ldots) .$$

Lemma5. Nechť K je temporální struktura a i je přirozené číslo. Pro pravdivostní hodnoty platí

$$\mathrm{K}^i_j(A) = \mathrm{K}_{i+j}(A)$$

pro každý index j a každou formuli A.

Důkaz provedeme indukcí podle složitosti formule A pro všechna j současně. Nechť

$$\mathrm{K} = (\eta_0, \eta_1, \eta_2, \ldots) \quad \text{a} \quad \mathrm{K}^i = (\eta_i, \eta_{i+1}, \eta_{i+2}, \ldots) .$$

Uvažujeme následující případy

(i) A je výroková konstanta $v \in V$. Potom

$$\mathrm{K}^i_j(v) = \eta_{i+j}(v) = \mathrm{K}_{i+j}(v) .$$

(ii) A je **false** , potom $K^i_j(\mathbf{false}) = \text{ff} = K_{i+j}(\mathbf{false})$.

(iii) A je $B \text{ -> } C$,

$$K^i_j(B \text{ -> } C) = \text{tt} \Leftrightarrow K^i_j(B) = \text{ff} \text{ nebo } K^i_j(C) = \text{tt}$$

$$\Leftrightarrow K_{i+j}(B) = \text{ff} \text{ nebo } K_{i+j}(C) = \text{tt}$$

$$\Leftrightarrow K_{i+j}(B \text{ -> } C) = \text{tt}$$

(iv) A je $\circ B$,

$$K^i_j(\circ B) = K^i_{j+1}(B)$$

$$= K_{i+j+1}(B)$$

$$= K_{i+j}(\circ B)$$

(v) A je $\Box B$,

$$\begin{aligned} \mathrm{K}^i_j(\Box B) &\Leftrightarrow \mathrm{K}^i_l(B) = \mathrm{tt} \quad \text{pro všechna } l \geq j \\ &\Leftrightarrow \mathrm{K}_{i+l}(B) = \mathrm{tt} \quad \text{pro všechna } l \geq j \\ &\Leftrightarrow \mathrm{K}_{i+j}(\Box B) \end{aligned}$$

Následující tvrzení je temporální obdobou Věty o dedukci v klasické výrokové logice.

Věta 6. $T \cup \{A\} \models B$ *právě když* $T \models \Box A \rightarrow B$.

Sémantický důkaz nebudeme provádět, ale povšimneme si, že z přesné obdoby klasické Věty o dedukci platí jen tvrzení

Věta 7. Je-li $T \models A \rightarrow B$ *potom* $T \cup \{A\} \models B$.

Obrácená implikace v LTL neplatí. Stačí uvažovat případ, kdy T je prázdná množina. Podle Věty 3 platí $A \models \Box A$ pro libovolnou formuli, ale implikace $A \rightarrow \Box A$ nemusí být validní formule. Není pravdivá v žádné temporální struktuře K, kde pro nějaké i a $j > i$ platí $\mathrm{K}_i(A) = \mathrm{tt}$ a $\mathrm{K}_j(A) = \mathrm{ff}$. Stačí uvažovat případ, kdy A je některá výroková konstanta.

Za zmínku stojí následující tvrzení

Věta. Je-li $T \models A$ a T je množina validních formulí potom $\models A$.

Důkaz. Nechť K je libovolná temporální struktura , v K je pravdivá každá validní formule, tedy K je modelem množiny T . Z předpokladu $T \models A$ dostáváme $K \models A$.

Protože K je libovolná temporální struktura, platí $\models A$ a A je validní formule .

Stejně jako v klasické logice je validita „duální“ pojem ke splnitelnosti.

Věta. $\models A$ *právě když* $\neg A$ není splnitelná .

Důkaz.

$\models A \Leftrightarrow \mathrm{K} \models A$ pro libovolnou strukturu K

$\Leftrightarrow \mathrm{K}_i(A) = \mathrm{tt}$ pro libovolné i

$\Leftrightarrow \mathrm{K}_i(\neg A) = \mathrm{ff}$ pro libovolné i

$\Leftrightarrow \mathrm{K} \not\models \neg A$

$\Leftrightarrow \neg A$ není splnitelná formule

Zde jsou některé často používané temporální formule a jejich neformální čtení.

A -> ○B *„jestliže A potom B v dalším stavu“*

A -> □ B *„jestliže A potom B teď a v každém dalším stavu“*

A -> ◊B *„jestliže A potom B teď nebo v nějakém dalším stavu“*

□(*A -> B*) *„jestliže A teď nebo v nějakém dalším stavu potom B platí ve stejném stavu “*

□○***A*** *„teď a za každým dalším stavem někdy platí A“*

„(od teď) A bude platit v nekonečně mnoha stavech“

○□***A*** *„od některého dalšího stavu bude stále platit A“*

„(od teď) A platí skoro vždycky“

Binární temporální operátory

Oblíbeným binárním operátorem je temporální obdoba programového konstruktu **until** . Nejčastěji se značí $A \cup B$.

Definice. ($A \cup B$ *a* A **B** B)

Pro temporální strukturu $\mathbf{K}$, index i a formule A , B definujeme pravdivostní hodnotu $\mathbf{K}_i (A \cup B)$ následovně

$\mathbf{K}_i (A \cup B) = \text{tt} \Leftrightarrow \mathbf{K}_j (B) = \text{tt}$ *pro nějaké* $j , j > i$ *a*
$\mathbf{K}_k (A) = \text{tt}$ *pro každé* $k , i < k < j$

.

Méně často se používá binární operátor A **B** B který čteme

„*A předchází B*“ nebo krátce „*A před B*“ ,

anglicky „*A before B*“ .

$\mathbf{K}_i\,(A\ \mathbf{B}\ B) = \mathrm{tt} \Leftrightarrow$ *pro každé* $j\,,\ j > i$ $\mathbf{K}_j\,(B) = \mathrm{tt}$ *implikuje*

$\mathbf{K}_k\,(A) = \mathrm{tt}$ *pro nějaké* $k\,,\ i < k < j$

jinak vyjádřeno

$$\mathbf{K}_i\,(A\ \mathbf{B}\ B) = \mathrm{tt} \Leftrightarrow (\forall j > i)(\mathbf{K}_j\,(B) = \mathrm{tt} \Rightarrow (\exists k)\,(i < k < j\ \&\ \mathbf{K}_k\,(A) = \mathrm{tt}))$$

Důležité validní formule

$\models \Box\neg A \leftrightarrow \neg \Diamond A$ $\qquad$ $\models \Diamond\neg A \leftrightarrow \neg \Box A$

(1) $\models \bigcirc\neg A \leftrightarrow \neg \bigcirc A$

$\models \Box \Diamond \neg A \leftrightarrow \neg \Diamond \Box A$ $\qquad$ $\models \Diamond \Box \neg A \leftrightarrow \neg \Box \Diamond A$

$\models ((\neg A) \cup B) \leftrightarrow \neg (A \mathbf{B} B)$

Následující validní implikace nelze zesílit na ekvivalence

.

$\models A \rightarrow \Diamond A$ $\qquad$ $\models \Box A \rightarrow A$

$\models \bigcirc A \rightarrow \Diamond A$ $\qquad$ $\models \Box A \rightarrow \bigcirc A$

$\models \Box A \rightarrow \Diamond A$ $\qquad$ $\models \Box A \rightarrow \bigcirc \Box A$

$\models A \cup B \rightarrow \Diamond A$ $\qquad$ $\models \Diamond \Box A \rightarrow \Box \Diamond A$

Idempotence ◊ , □ , □◊ a ◊□

$\models \Diamond\Diamond A \leftrightarrow \Diamond A$ $\models \Box\Box A \leftrightarrow \Box A$

$\models \Diamond\Box\ \Diamond\Box\ A \leftrightarrow \Diamond\Box\ A$ $\models \Box\Diamond\ \Box\Diamond\ A \leftrightarrow \Box\Diamond\ A$

Ale operátor příštího stavu není idempotentní. Formule $\bigcirc\bigcirc A \leftrightarrow \bigcirc A$ není validní.

Nekonečné modality □◊ a ◊□ „konzumují“ všechny ostatní modality s jedním argumentem, které jsou na ně aplikované. S menším násilím na syntax formulí to můžeme kompaktně vyjádřit takto

$\models (\Box\Diamond)\ A \leftrightarrow \bigcirc\ (\Box\Diamond)\ A \leftrightarrow \Diamond\ (\Box\Diamond)\ A \leftrightarrow \Box\ (\Box\Diamond)\ A$

. $\models \Box\Diamond\ (\Box\Diamond)\ A \leftrightarrow \Diamond\Box\ (\Box\Diamond)\ A$

. $\models (\Diamond\Box)\ A \leftrightarrow \bigcirc\ (\Diamond\Box)\ A \leftrightarrow \Diamond\ (\Diamond\Box)\ A \leftrightarrow \Box\ (\Diamond\Box)\ A$

. $\models \Box\Diamond\ (\Diamond\Box)\ A \leftrightarrow \Diamond\Box\ (\Diamond\Box)\ A$

Podle své definice jsou operátory $\Diamond$ a $\Box\Diamond$ povahy existenční a operátory $\Box$ a $\Diamond\Box$ povahy univerzální, zatímco operátor $\cup$ (*until*) je v prvním argumentu univerzální a ve druhém argumentu existenční.
Vykazují podobné chování jako existenční kvantifikátor a univerzální kvantifikátor v predikátové logice.

$\models \Diamond (A \vee B) \leftrightarrow (\Diamond A \vee \Diamond B)$ $\quad$ $\models \Box\Diamond (A \vee B) \leftrightarrow (\Box\Diamond A \vee \Box\Diamond B)$

$\models \Box (A \wedge B) \leftrightarrow (\Box A \wedge \Box B)$ $\quad$ $\models \Diamond\Box (A \wedge B) \leftrightarrow (\Diamond\Box A \wedge \Diamond\Box B)$

Pro operátor $\cup$ (*until*) a boolovské spojky konjunkce a disjunkce platí tyto distributivní vztahy.

$\models ((A \wedge B) \cup C) \leftrightarrow ((A \cup C) \wedge (B \cup C))$

$\models (A \cup (B \vee C)) \leftrightarrow ((A \cup C) \vee (B \cup C))$

Operátor ○ (next) se vztahuje k jedinému časovému bodu, proto se distribuuje se všemi boolovskými spojkami.

(2) $\models \bigcirc(A \vee B) \leftrightarrow (\bigcirc A \vee \bigcirc B)$ $\models \bigcirc(A \wedge B) \leftrightarrow (\bigcirc A \wedge \bigcirc B)$
$\models \bigcirc(A \rightarrow B) \leftrightarrow (\bigcirc A \rightarrow \bigcirc B)$ $\models \bigcirc(A \leftrightarrow B) \leftrightarrow (\bigcirc A \leftrightarrow \bigcirc B)$

přitom ekvivalence $\models \bigcirc\neg A \leftrightarrow \neg \bigcirc A$ již byla uvedena výše.

Pro kombinace operátorů universální a existenční povahy platí jen implikace, které nelze zesílit na ekvivalence.

$\models (\Box A \vee \Box B) \rightarrow \Box (A \vee B)$ $\models \Diamond\Box (A \vee B) \rightarrow (\Diamond\Box A \vee \Diamond\Box B)$
$\models \Diamond (A \wedge B) \rightarrow (\Diamond A \wedge \Diamond B)$ $\models \Box\Diamond (A \wedge B) \rightarrow (\Box\Diamond A \wedge \Box\Diamond B)$

$\models ((A \cup C) \vee (B \cup C)) \rightarrow ((A \vee B) \cup C)$.
$\models (A \cup (B \wedge C)) \rightarrow ((A \cup B) \wedge (A \cup C))$

Povšimneme si, že uvedené operátory jsou monotonní v každém argumentu.

$\models \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$ $\quad\quad$ $\models \Box(A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$

$\models \Box(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$

$\models \Box(A \rightarrow B) \rightarrow (\Diamond\Box A \rightarrow \Diamond\Box B)$ $\quad\quad$ $\models \Box(A \rightarrow B) \rightarrow (\Diamond\Box A \rightarrow \Diamond\Box B)$

$\models \Box(A \rightarrow B) \rightarrow ((A \cup C) \rightarrow (B \cup C))$

$\models \Box(A \rightarrow B) \rightarrow ((C \cup A) \rightarrow (C \cup B))$

Nakonec uvedeme důležité charakteristiky temporálních operátorů pomocí pevných bodů.

$\models \Diamond A \leftrightarrow A \vee \bigcirc\Diamond A$ $\quad\quad$ (3) $\quad\quad$ $\models \Box A \leftrightarrow A \wedge \bigcirc\Box A$

$\models (A \cup B) \leftrightarrow B \vee (A \wedge \bigcirc(A \cup B))$

$\models (A \mathbf{B} B) \leftrightarrow \neg B \wedge (A \vee \bigcirc(A \mathbf{B} B))$

Temporální logika

Formální systém

Formální systém

Výroková lineární temporální logika (LTL) v sobě zahrnuje klasickou výrokovou logiku v podobném smyslu jako predikátová logika. Tautologie výrokové logiky se transformovaly na validní formule predikátové logiky tím, že se za výrokové proměnné (konzistentně) dosadily predikátové formule. Stejný postup můžeme použít i pro výrokovou temporální logiku.

Například. Výroková formule

$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$$

kde p, q, r jsou výrokové proměnné, je tautologie, tedy validní formule klasické výrokové logiky. Dosazením temporálních formulí $\circ A$, $\Box B$, $\Diamond C$ za výrokové proměnné p , q , r vznikne temporální formule

$$((\circ A \rightarrow \Box B) \wedge (\Box B \rightarrow \Diamond C)) \rightarrow (\circ A \rightarrow \Diamond C)$$

která je validní v temporální logice.

Definice. (Tautologicky validní formule)

Říkáme, že temporální formule $\boldsymbol{A}$ je tautologicky (výrokově) validní, jestliže $\boldsymbol{A}$ vznikne z nějaké tautologie výrokové logiky $\boldsymbol{A}^*(p_1, p_2, \ldots, p_n)$, kde všechny výrokové proměnné ve formuli $\boldsymbol{A}^*$ jsou uvedeny, nahrazením výrokových proměnných $p_1, p_2, \ldots, p_n$ po řadě temporálními formulemi $A_1, A_2, \ldots, A_n$.

Věta 1.

Každá tautologicky validní formule je validní.

Důkaz. Předpokládejme, že $\boldsymbol{A}^*$ je tautologie a že temporální formule $\boldsymbol{A}$ vznikla z $\boldsymbol{A}^*$ nahrazením výrokových proměnných jako v předchozí definici. Máme ukázat, že pro každou temporální strukturu $\mathbf{K}$ a každý časový bod i, $i \geq 0$ platí $\mathbf{K}_i(\boldsymbol{A}) = \mathrm{tt}$.

Nechť v je libovolné pravdivostní ohodnocení v (klasické) výrokové logice výrokových proměnných ve formuli $\boldsymbol{A}^*$. Protože $\boldsymbol{A}^*$ je tautologie, $\boldsymbol{A}^*$ je pravdivá nezávisle na pravdivostních hodnotách $v(p_j)$ výrokových proměnných p_j, $j = 1, 2, \ldots, n$.

V každém časovém bodě i, $i \geq 0$ temporální struktura $\mathbf{K}$ určuje pravdivostní hodnoty $\mathrm{K}_i(A_j)$ j, $j = 1, 2, \ldots, n$. Protože sémantika pro výrokové spojky v temporální logice je definována stejným způsobem jako v klasické výrokové logice, pravdivostní hodnota $\mathrm{K}_i(A) = \mathrm{tt}$ bez ohledu na pravdivostní hodnoty $\mathrm{K}_i(A_j)$ podformulí A_j.

Protože $\boldsymbol{A}$ je pravdivá v každém časovém bodě i, je pravdivá i ve struktuře $\mathbf{K}$ a protože $\mathbf{K}$ je libovolná temporální struktura, formule $\boldsymbol{A}$ je validní.

Z věty 1 také vyplývá, že převod tautologií klasické výrokové logiky do LTL je možné rozšířit i na relaci sémantického důsledku $\models$.

Předpokládejme, že formule $\boldsymbol{B}$ je důsledkem množiny formulí $\boldsymbol{T}$ ve výrokové logice. Jestliže konzistentně dosadíme temporální formule do výrokových formulí z $\boldsymbol{T}$ a do formule $\boldsymbol{B}$ neměli bychom poškodit relaci $\models$.

Například

$$\bigcirc \boldsymbol{D} \rightarrow \Box \boldsymbol{E} \ , \quad \Box \boldsymbol{E} \rightarrow \Diamond \boldsymbol{F} \ \models \ \bigcirc \boldsymbol{D} \rightarrow \Diamond \boldsymbol{F}$$

by mělo platit protože ve výrokové logice platí

$$\boldsymbol{A} \rightarrow \boldsymbol{B} \ , \boldsymbol{B} \rightarrow \boldsymbol{C} \ \models \ \boldsymbol{A} \rightarrow \boldsymbol{C}$$

Pro jednoduché vyjádření tohoto jevu si připomeňme, že ve výrokové logice platí

$$\boldsymbol{A}_1 , \boldsymbol{A}_2 , \ldots , \boldsymbol{A}_n \models \boldsymbol{B} \quad \text{právě když} \quad \models \boldsymbol{A}_1 \rightarrow (\boldsymbol{A}_2 \rightarrow \ldots \rightarrow (\boldsymbol{A}_n \rightarrow \boldsymbol{B})$$

Tím je motivována následující definice.

Definice. (Tautologický důsledek konečné množiny formulí)

Nechť $A_1, A_2, \ldots, A_n$, $n \geq 0$ a B jsou temporální formule. Říkáme, že formule B je tautologickým důsledkem formulí $A_1, A_2, \ldots, A_n$, jestliže formule

$$A_1 \text{ -> } (A_2 \text{ -> } \ldots \text{ -> } (A_n \text{ -> } B))$$

je tautologicky validní.

Věta 2.

Je-li B tautologickým důsledkem temporálních formulí $A_1, A_2, \ldots, A_n$, potom $A_1, A_2, \ldots, A_n \models B$.

Důkaz. Podle předpokladu je

$$\boldsymbol{A}_1 \text{ -> } (\boldsymbol{A}_2 \text{ -> } \ldots \text{ -> } (\boldsymbol{A}_n \text{ -> } \boldsymbol{B})$$

tautologicky validní formule. Podle Věty 1 to znamená, že je to validní formule, tedy

$$\models \boldsymbol{A}_1 \text{ -> } (\boldsymbol{A}_2 \text{ -> } \ldots \text{ -> } (\boldsymbol{A}_n \text{ -> } \boldsymbol{B})$$

Použijeme-li *n* krát Větu 7 ze sémantiky LTL, dostaneme tvrzení věty.

Formální systém Lineární temporální logiky.

Axiomy.

(Taut) Všechny tautologicky validní formule

Pro libovolné formule A , B jsou následující formule axiomy

(LTL 1) $\neg\circ A \leftrightarrow \circ\neg A$

(LTL 2) $\circ(A \rightarrow B) \rightarrow (\circ A \rightarrow \circ B)$

(LTL 3) $\Box A \rightarrow (A \wedge \circ\Box A)$

Odvozovací pravidla.

(MP) $A\ ,\ A \rightarrow B\ \vdash\ B$

(Next) $A\ \vdash\ \circ A$

(Indukce) $A \rightarrow B\ ,\ A \rightarrow \circ A \vdash\ A \rightarrow \Box B$

Axiom (taut) je schema, kde bychom spíše očekávali instance axiomů klasické výrokové logiky. Pro nás není důležité jak se odvozují tautologicky pravdivé formule ; to probíhá v stejně jako v klasické výrokové logice. Abychom zkrátili důkazy o tuto „klasickou část“ , bereme všechny tautologicky pravdivé formule jako axiomy. Množina všech tautologicky pravdivých formulí je rozhodnutelná, tedy existuje algoritmus, který rozpoznává tautologicky pravdivé formule (respektive jejich kódy).

Axiomy LTL2 a LTL3 jsou implikace a ne ekvivalence i když obrácené implikace jsou validní. Ukážeme, že tyto implikace jsou dokazatelné.

Odvozovací pravidlo (ind) je pravidlo indukce, které lze neformálně vyslovit takto

„jestliže A (vždycky) implikuje B a A je invariantní při přechodu z libovolného stavu do do následujícího, potom vždy platí B “.

Ukážeme, že formální systém LTL je korektní vzhledem k sémantice.

Věta 3. (o korektnosti pro LTL)

Nechť A je formule a T je množina formulí, potom

$T \vdash A$ implikuje $T \models A$ speciálně $\vdash A$ implikuje $\models A$.

Důkaz. Postupujeme indukcí podle důkazu formule.

(a) Je-li A axiom ze schemat (taut), LTL1, LTL2, LTL3, potom podle Věty 1 jsou všechny axiomy z (taut) , tedy všechny tautologicky validní formule validní.
Axiom ze schematu LTL1 je validní formule, podle sémantické ekvivalence (1) z prezentace Temporální logika 1. Axiomy ze schemat LTL2 a LTL3 jsou validní podle sémantických ekvivalencí (2) a (3) z předchozí prezentace.

(b) Je-li A prvkem T, potom $T \models A$ plyne z definice modelu.

(c) Je-li A odvozena pravidlem modus ponens z fomulí B a $B \rightarrow A$, podle indukčního předpokladu je $T \models B$ a $T \models B \rightarrow A$. Ze sémantické korektnosti pravidla modus ponens dostáváme $T \models A$.

(d) Je-li A odvozena pravidlem (nex), tedy A je tvaru $\circ B$, takové že $T \vdash B$, z indukčního předpokladu dostáváme $T \models B$ a z věty 4 z předchozí prezentace dostáváme $T \models B \rightarrow \circ B$ odkud $T \models A$ ze sémantické korektnosti pravidla modus ponens.

(e) Je-li A odvozena z formulí $B \rightarrow C$ a $B \rightarrow \circ B$ pravidlem (ind), potom A je tvaru $B \rightarrow \Box C$. Z indukčního předpokladu dostáváme $T \models B \rightarrow C$ a $T \models B \rightarrow \circ B$ a z věty 4 z předchozí prezentace dostáváme $T \models B \rightarrow \Box C$, tedy $T \models A$.

I když při formulaci schematu axiomů (taut) jsme dali přednost tautologicky validním formulím před instancemi axiomů klasické výrokové logiky, přesto je ve výrokové temporální logice (LTL) čistě klasický fragment, který používá jen axiomy ze schematu (taut) a z nich odvozuje jen pomocí pravidla modus ponens.
Dokazatelnost v tomto smyslu budeme vyjadřovat pomocí následujícího odvozeného pravidla, které je vlastně zobecněním pravidla modus ponens

(prop) $A_1 , A_2 , \ldots , A_n \ |- B$ jestliže
B je tautologickým důsledkem formulí $A_1 , A_2 , \ldots , A_n$

Příkladem může být pravidlo skládání implikací

$$A \rightarrow B , B \rightarrow C \ |- \ A \rightarrow C$$

které budeme používat v důkazech stejně jako mnoho dalších jako pravidlo typu (prop). Tento postup je odůvodněn následující větou.

Věta 4.

$A_1 , A_2 , \ldots , A_n \vdash B$ jestliže

. B je tautologickým důsledkem formulí $A_1 , A_2 , \ldots , A_n$

Důkaz. Dokážeme jen pro $n = 2$, obecný případ se dokazuje podobně.

Je-li B tautologickým důsledkem A_1 , A_2 , potom formule

$$A_1 \rightarrow (A_2 \rightarrow B)$$

je tautologicky validní a je možné sestrojit následující důkaz formule B z formulí A_1 , A_2 :

(1) A_1	předpoklad
(2) A_2	předpoklad
(3) $A_1 \rightarrow (A_2 \rightarrow B)$	(taut)
(4) $(A_2 \rightarrow B)$	(1),(3) (mp)
(5) B	(2),(4) (mp)

Příklady dalších důkazů

Nejprve obrácené implikace k axiomům (LTL2) a (LTL3). Tím bude dokončen formální důkaz validních formulí (2), (3) ze sémantiky LTL.

(LTL2′) $\circ(A \rightarrow B) \leftarrow (\circ A \rightarrow \circ B)$

(LTL3′) $\Box A \leftarrow A \wedge \circ\Box A$

Důkaz (LTL2′)

(1) $\neg(A \rightarrow B) \rightarrow A$	(taut)
(2) $\circ(\neg(A \rightarrow B) \rightarrow A)$	(nex), (1)
(3) $\circ(\neg(A \rightarrow B) \rightarrow A) \rightarrow (\circ\neg(A \rightarrow B) \rightarrow \circ A)$	(LTL2)
(4) $(\circ\neg(A \rightarrow B) \rightarrow \circ A)$	(mp), (2), (3)
(5) $\neg\circ(A \rightarrow B) \leftrightarrow \circ\neg(A \rightarrow B)$	(LTL1)

(6) $\circ\neg(A \rightarrow B) \rightarrow \circ A$	(prop),(4),(5)
(7) $\neg (A \rightarrow B) \rightarrow \neg B$	(taut)
(8) $\circ\neg(A \rightarrow B) \rightarrow \circ\neg B$	jako (6) z (1)
(9) $\circ\neg(B) \rightarrow \neg\circ B$	(prop),(LTL1)
(10) $\circ\neg(A \rightarrow B) \rightarrow \neg\circ B$	(prop),(8),(9)
(11) $\circ\neg(A \rightarrow B) \rightarrow \neg(\circ A \rightarrow \circ B)$	(prop),(6),(10)
(12) $(\circ A \rightarrow \circ B) \rightarrow \circ (A \rightarrow B)$	(prop),(11)

Důkaz (LTL3’)

(1) $(A \wedge \bigcirc\Box A) \rightarrow A$	(taut)
(2) $\Box A \rightarrow (A \wedge \bigcirc\Box A)$	(LTL3)
(3) $\bigcirc(\Box A \rightarrow (A \wedge \bigcirc\Box A))$	(nex),(2)
(4) $\bigcirc\Box A \rightarrow \bigcirc(A \wedge \bigcirc\Box A)$	(mp),(LTL2),(3)
(5) $(A \wedge \bigcirc\Box A) \rightarrow \bigcirc(A \wedge \bigcirc\Box A)$	(prop),(4)
(6) $(A \wedge \bigcirc\Box A) \rightarrow \Box A$	(ind),(1),(5)

Následující odvozovací pravidla jsou varianty indukčního pravidla.

(ind1) $A \rightarrow \circ A \vdash A \rightarrow \Box A$

(ind2) $A \rightarrow B \quad B \rightarrow \circ B \vdash A \rightarrow \Box B$

Odvození (ind1)

(1) $A \rightarrow \circ A$	předpoklad
(2) $A \rightarrow A$	(taut)
(3) $A \rightarrow \Box A$	(ind),(1),(2)

Odvození (ind2)

(1) $\boldsymbol{A} \rightarrow \boldsymbol{B}$	předpoklad
(2) $\boldsymbol{B} \rightarrow \circ\boldsymbol{B}$	předpoklad
(3) $\boldsymbol{B} \rightarrow \Box\boldsymbol{B}$	(ind),(1),(2)
(4) $\boldsymbol{A} \rightarrow \Box\boldsymbol{B}$	(prop),(1),(3)

Následují dvě odvozovací pravidla. První je obdobou (nex) pro □.

(alw) $A \vdash \Box A$

(som) $A \rightarrow \circ B \vdash A \rightarrow \Diamond B$

Odvození (alw)

(1) A	předpoklad
(2) $\circ A$	(nex),(1)
(3) $A \rightarrow \circ A$	(prop),(2)
(4) $A \rightarrow \Box A$	(ind1),(3)
(5) $\Box A$	(mp),(1),(4)

Odvození (som)

(1) $\boldsymbol{A}$ -> ○$\boldsymbol{B}$	předpoklady , (prop)
(2) □¬$\boldsymbol{B}$ -> (¬$\boldsymbol{B}$ ∧○□ ¬$\boldsymbol{B}$)	(LTL3)
(3) □¬$\boldsymbol{B}$ -> ¬$\boldsymbol{B}$	(prop),(2)
(4) □¬$\boldsymbol{B}$ -> ○□ ¬$\boldsymbol{B}$	(prop),(2)
(5) ○(□¬$\boldsymbol{B}$ -> ¬$\boldsymbol{B}$)	(nex),(3)
(6) ○(□¬$\boldsymbol{B}$ -> ¬$\boldsymbol{B}$) -> (○□ ¬$\boldsymbol{B}$ -> ○¬$\boldsymbol{B}$)	(LTL2)
(7) ○□ ¬$\boldsymbol{B}$ -> ○¬$\boldsymbol{B}$	(mp),(5),(6)
(8) □ ¬$\boldsymbol{B}$ -> ○¬$\boldsymbol{B}$	(prop),(4),(7)
(9) ¬○$\boldsymbol{B}$ <-> ○¬$\boldsymbol{B}$	(LTL1)
(10) □ ¬$\boldsymbol{B}$ -> ○¬$\boldsymbol{B}$	(prop),(8),(9)
(11) ○$\boldsymbol{B}$ -> ¬□ ¬$\boldsymbol{B}$	(prop),(10)
(12) $\boldsymbol{A}$ -> ◊$\boldsymbol{B}$	(prop),(1),(11)

Lemma 5.

$$(\circ A \wedge \circ B) \;\text{->}\; \circ(A \wedge B)$$

Důkaz.

(1) $\circ(A \text{->} \neg B) \text{->} (\circ A \text{->} \circ\neg B)$	(LTL2)
(2) $\circ(A \text{->} \neg B) \text{->} (\circ A \text{->} \neg\circ B)$	(prop),(LTL1),(1)
(3) $\neg(\circ A \text{->} \circ\neg B) \text{->} \neg\circ (A \text{->} \neg B)$	(prop),(2)
(4) $\neg(\circ A \text{->} \circ\neg B) \text{->} \circ\neg(A \text{->} \neg B)$	(prop),(LTL1),(3)
(5) $(\circ A \wedge \circ B) \text{->} \circ(A \wedge B)$	(prop)

Věta 6. (o dedukci pro LTL)

Nechť A, B jsou formule a T je množina formulí. Platí

$$T \cup \{A\} \vdash B \quad \text{právě když} \quad T \vdash \Box A \rightarrow B$$

Důkaz (indukcí podle odvození/délky důkazu)

$\Rightarrow$ (a) Je-li B axiom LTL nebo B je prvkem T, potom $T \vdash B$ a pomocí (prop) dostáváme $T \vdash \Box A \rightarrow B$.

(b) je-li B formule A, potom $T \vdash \Box A \rightarrow A \wedge \bigcirc \Box A$ podle LTL3 a $T \vdash \Box A \rightarrow A$ lze odvodit pomocí (prop).

(c) Je-li B odvozena pravidlem (mp) z formulí C, $C \rightarrow B$.
Potom

$$T \cup \{A\} \vdash C \quad \text{a} \quad T \cup \{A\} \vdash C \rightarrow B$$

a z indukčního předpokladu

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow \boldsymbol{C} \quad \text{a} \quad \boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow (\boldsymbol{B} \rightarrow \boldsymbol{C})$$

(d) Je-li $\boldsymbol{B}$ tvaru $\circ\boldsymbol{C}$ a byla odvozena pravidlem (nex) z formule $\boldsymbol{C}$, potom

$$\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \boldsymbol{C}$$

a z indukčního předpokladu

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow \boldsymbol{C}$$

zbývá dokázat

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow \circ\boldsymbol{C}$$

(1)	$\Box\boldsymbol{A} \rightarrow \boldsymbol{C}$	již odvozeno
(2)	$\circ(\Box\boldsymbol{A} \rightarrow \boldsymbol{C})$	(nex),(1)
(3)	$\circ(\Box\boldsymbol{A} \rightarrow \boldsymbol{C}) \rightarrow (\circ\Box\boldsymbol{A} \rightarrow \circ\boldsymbol{C})$	(LTL2)
(4)	$\circ\Box\boldsymbol{A} \rightarrow \circ\boldsymbol{C}$	(mp),(2),(3)
(5)	$\Box\boldsymbol{A} \rightarrow (\boldsymbol{A} \wedge \circ\Box\boldsymbol{A})$	(LTL3)
(6)	$\Box\boldsymbol{A} \rightarrow \circ\Box\boldsymbol{A}$	(prop),(5)
(7)	$\Box\boldsymbol{A} \rightarrow \circ\boldsymbol{C}$	(prop),(4),(6)

(e) $\boldsymbol{B}$ je $\boldsymbol{C} \rightarrow \Box\boldsymbol{D}$. a je odvozeno pravidlem (ind) z formulí

$$\boldsymbol{C} \rightarrow \boldsymbol{D} \quad \text{a} \quad \boldsymbol{C} \rightarrow \circ\boldsymbol{C}$$

potom z indukčního předpokladu

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \boldsymbol{D}) \qquad \boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \circ\boldsymbol{C})$$

odtud odvodíme

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \Box\boldsymbol{D}) \qquad \text{pomocí lemmatu 5.}$$

(1) $\Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \boldsymbol{D})$	odvozeno
(2) $\Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \circ\boldsymbol{C})$	odvozeno
(3) $(\Box\boldsymbol{A} \wedge \boldsymbol{C}) \rightarrow \boldsymbol{D}$	(prop),(1)
(4) $(\Box\boldsymbol{A} \wedge \boldsymbol{C}) \rightarrow \circ\boldsymbol{C}$	(prop),(2)
(5) $\Box\boldsymbol{A} \rightarrow \circ\Box\boldsymbol{A}$	(LTL3),(prop)
(6) $(\Box\boldsymbol{A} \wedge \boldsymbol{C}) \rightarrow (\circ\Box\boldsymbol{A} \wedge \circ\boldsymbol{C})$	(prop),(4),(5)
(7) $(\circ\Box\boldsymbol{A} \wedge \circ\boldsymbol{C}) \rightarrow \circ(\Box\boldsymbol{A} \wedge \boldsymbol{C})$	lemma 5
(8) $(\Box\boldsymbol{A} \wedge \boldsymbol{C}) \rightarrow \circ(\Box\boldsymbol{A} \wedge \boldsymbol{C})$	(prop),(6),(7)
(9) $(\Box\boldsymbol{A} \wedge \boldsymbol{C}) \rightarrow \Box\boldsymbol{D}$	(ind),(3),(8)
(10) $\Box\boldsymbol{A} \rightarrow (\boldsymbol{C} \rightarrow \Box\boldsymbol{D})$	(prop),(9)

Tím je implikace zleva doprava dokázána. Opačná implikace má kratší důkaz.

<= Předpokládejme, že

$$\boldsymbol{T} \vdash \Box \boldsymbol{A} \rightarrow \boldsymbol{B}$$

potom také

$$\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \Box \boldsymbol{A} \rightarrow \boldsymbol{B}$$

při tom

$$\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \Box \boldsymbol{A}$$

podle (alw), odkud

$$\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \boldsymbol{B}$$

plyne pomocí (mp).

Poznámka. Z Věty o dedukci klasické výrokové logiky se stejným způsobem odvodí implikace

Je-li $\boldsymbol{T} \vdash \boldsymbol{A} \rightarrow \boldsymbol{B}$ potom $\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \boldsymbol{B}$ .Opačná implikace není v LTL dokazatelná.

V poznámce k sémantické verzi klasické Věty o dedukci se ukazuje, že z

$$\boldsymbol{T} \cup \{\boldsymbol{A}\} \models \Box\boldsymbol{A}$$

neplyne

$$\boldsymbol{T} \models \boldsymbol{A} \rightarrow \Box\boldsymbol{A}$$

podle Věty o korektnosti pro LTL pak tvrzení

Je-li $\boldsymbol{T} \cup \{\boldsymbol{A}\} \vdash \Box\boldsymbol{A}$ potom $\boldsymbol{T} \vdash \boldsymbol{A} \rightarrow \Box\boldsymbol{A}$

není v LTL dokazatelné.

Úplnost LTL

Uvažujme množinu formulí

$$\boldsymbol{T} = \{\boldsymbol{A}, \bigcirc\boldsymbol{A}, \bigcirc\bigcirc\boldsymbol{A}, \bigcirc\bigcirc\bigcirc\boldsymbol{A}, \ldots, \bigcirc^{n}\boldsymbol{A}, \bigcirc^{n+1}\boldsymbol{A}, \ldots\}$$

Snadno se odvodí

$$\boldsymbol{T} \models \Box\boldsymbol{A} \qquad (1)$$

ale to neplatí pro žádnou konečnou podmnožinu $\boldsymbol{T'}$ množiny $\boldsymbol{T}$. Nemůžeme tedy očekávat, že v LTL bude možné odvodit

$$\boldsymbol{T} \vdash \Box\boldsymbol{A} \qquad (1')$$

V takovém důkazu by bylo použito jen konečně mnoho formulí z $\boldsymbol{T}$ a to znamená, že by platilo

$$\boldsymbol{T'} \vdash \Box\boldsymbol{A}$$

pro nějakou konečnou podmnožinu $\boldsymbol{T'}$ množiny $\boldsymbol{T}$.

Z Věty o korektnosti pro LTL by plynulo

$$T' \models \Box A$$

a to není možné. V LTL tedy neplatí tvrzení:

Je-li $T \models A$ potom $T \vdash A$

pro každou množinu T a každou formuli A. Dá se ukázat, že v LTL platí jen slabá forma Věty o úplnosti.

Věta 7. (Slabá Věta o úplnosti pro LTL)

Pro každou konečnou množinu formulí T a libovolnou formuli A platí tvrzení: Je-li

$T \models A$ potom $T \vdash A$,

a speciálně $\models A$ potom $\vdash A$.

Důsledek 8.

Všechny validní formule jsou dokazatelné v LTL .

Speciálně všechny validní formule, které jsme zmínili při studiu sémantiky jsou v LTL dokazatelné.

Důkaz. Použijeme-li ještě Větu o korektnosti dostaneme v LTL tvrzení:

$$\vdash A \quad \text{právě když} \quad \models A \ .$$

Důsledek 9.

Pro každou konečnou množinu formulí T a libovolnou formuli A platí tvrzení:

. $$T \vdash A \quad \text{právě když} \quad T \models A \ .$$

Aplikace Temporální logiky

V temporální logice je možné definovat vlastnosti Stavových systémů, ověřovat je a odvozovat z nich další důsledky. Termín *stavový systém* budeme používat intuitivně pro nějaký systém Γ, který v jednotlivých krocích prochází různými stavy a vykazuje určité „chování“.

Každý průchod systému Γ množinou stavů (tzv. exekuční posloupnost) budeme krátce nazývat *běh* .

Každá temporální formule v jazyce $L_{LTL\,\Gamma}$ systému Γ specifikuje nějakou vlastnost stavového systému Γ, kterou můžeme ztotožnit s určitou množinou běhů systému Γ. Můžeme říci, že každá temporální formule určuje nějakou množinu běhů.

Ale neplatí to naopak. Pokud Γ má nekonečnou množinu běhů, potom podmnožin množiny všech běhů je nespočetně mnoho a naproti tomu je jen spočetně mnoho temporálních formulí. To znamená, že ne každá množina běhů může být definována nějakou temporální formulí.

Tento triviální fakt musíme vzít na vědomi. Je však zřejmé, že to jaké vlastnosti systému Γ budeme moci specifikovat formulemi temporální logiky záleží na volbě jazyka $L_{LTL\,\Gamma}$.

Různé „deskripční jazyky“ pro popis systému dovolují první hrubou klasifikaci vlastností systému.

Můžeme to ukázat na jednoduchém systému $\Gamma_{\alpha\beta}$, který má jenom dvě akce α, β takové, že výrokové konstanty $exec\,\alpha$, $exec\,\beta$ nemohou být současně pravdivé ve stejném stavu.

Do jazyka $L_{LTL\,\Gamma}$ přidáme ještě jeden temporální operátor $<>$, který se (proti zavedenému zvyku) vztahuje k minulým stavům.

Pro temporální strukturu $\mathbf{K}$, přirozené číslo i a formuli A, která neobsahuje operátor $<>$ budeme definovat pravdivostní hodnotu

$\mathbf{K}_i(<>A)$ následujícím způsobem

$\mathbf{K}_i(<>A) =$ tt jestliže $\mathbf{K}_j(A) =$ tt pro nějaké j, $j < i$

$\mathbf{K}_i(<>A) =$ ff jestliže $\mathbf{K}_j(A) =$ ff pro každé j, $j < i$

(včetně případu, kdy takové j neexistuje)

V takto rozšířeném jazyce budeme říkat, že A je retro-formule, pokud obsahuje jen retro-operátor $<>$.

Formulím tvaru $\odot A$, kde $\odot$ je některý z pozitivních operátorů $\Box$, $\bigcirc$, $\Diamond$, $\Diamond\Box$, $\Box\Diamond$ a A je retro budeme říkat *precedenční*.

Nyní můžeme klasifikovat vlastnosti, které lze vyjádřit pomocí precedenčních formulí. Jako příklady použijeme formule z $\Gamma_{\alpha\beta}$.

(a) Vlastnost specifikovaná precedenční formulí

$$\Box A$$

kde A je retro, nazýváme *bezpečnostní*. Příkladem je formule

$$\Box(exec\ \beta \ \text{->}\ \text{<>}exec\ \alpha)$$

kterou čteme

„kdykoliv je provedena akce β , α bylo provedeno někdy před tím“.

(b) Vlastnost specifikovaná precedenční formulí

$$\Diamond A$$

kde A je retro, nazveme *odpovědní.* Příkladem je formule

$$\Diamond(<>exec\ \alpha \ \text{->}\ exec\ \beta)$$

„ někdy později bude na dřívější provedení akce α odpovězeno akcí β“

(c) Vlastnost specifikovaná precedenční formulí

$$\Diamond\Box A$$

kde A je retro, nazveme *perzistentní* . Příkladem je formule

$$\Diamond\Box \neg exec\ \alpha$$

„od některého budoucího stavu, akce α již nebude nikdy provedena“

Čtvrtý terminální prefix $\Box\Diamond$ nedává novou třídu vlastností, protože formule $\Box\Diamond A$ je sémanticky ekvivalentní s formulí $\Box A$. To znamená, že vlastnosti $\Box\Diamond A$ pro retro-formule A jsou odpovědní vlastnosti .

Ostatně tři třídy vlastností (a), (b), (c) nejsou disjunktní. Každá bezpečnostní vlastnost je také odpovědní a perzistentní vlastnost.

Když použijeme binární operátory, formule tvaru

$$A \rightarrow (B \ \bigcirc \ C)$$

kde $\bigcirc$ je některý z operátorů **before** , **after , atnext , unless** atd., kde A , B a C jsou stavové formule daného systému Γ budeme také nazývat *precedenční*.

Binární operátory **before** , **after , atnext** mají celkem názorný intuitivní význam. Operátor **unless** je definován podobně jako **until** ale jeho pravdivost závisí na více stavech.

Pro temporální strukturu **K** , formule A , B přirozené číslo i definujeme

$\mathbf{K}_i(A \textbf{ unless } B) = \text{tt} \Leftrightarrow \mathbf{K}_i(B) = \text{tt}$ pro nějaké $j > i$ a
$(\mathbf{K}_k(A) = \text{tt}$ pro každé k , $i < k < j$ nebo
$\mathbf{K}_k(A) = \text{tt}$ pro každé k , $k > j$)

$\mathbf{K}_i(A \textbf{ unl } B) = \text{tt} \Leftrightarrow \mathbf{K}_i(B) = \text{tt}$ pro nějaké $j \geq i$ a
$(\mathbf{K}_k(A) = \text{tt}$ pro každé k , $i < k < j$
nebo
$\mathbf{K}_k(A) = \text{tt}$ pro každé k , $k > j$)

Precedenční vlastnosti specifikované pomocí těchto binárních operátorů jsou speciálním případem bezpečnostních vlastností: například platí

$$A \text{ -> } (B \textbf{ before } C) \;\Leftrightarrow\; \Box(C \text{ -> } (B \textbf{ after } A))$$

Klasifikace vlastností systémů by mohla pokračovat pokud bychom použili jazyky s jemnějšími (podrobnějšími) výrazovými prostředky.

Opustíme tuto linku a v dalším se budeme věnovat některým jednoduchým případům tříd vlastností, které popíšeme podrobněji.

Zavedeme a popíšeme základní metody verifikace vlastností pro již zmíněné tři typy vlastností.

Metody verifikace závisí na tom, jak stavovému systému rozumíme a jak jeho relevantní vlastnosti můžeme popsat pomocí formálního přechodového systému, který stavový systém reprezentuje.

a) Invariantní vlastnosti.

Chceme-li odvodit invariantní vlastnost

$$A \rightarrow \Box B$$

Nejčastěji se metoda důkazu opírá o použití některého z indukčních odvozovacích pravidel:

(ind) $A \rightarrow B,\ A \rightarrow \circ A \vdash A \rightarrow \Box B$

(ind1) $A \rightarrow \circ A \vdash A \rightarrow \Box A$

(ind2) $A \rightarrow B,\ B \rightarrow \circ B \vdash A \rightarrow \Box B$

Z nich například pravidlo (ind1) lze použít k odvození formule bezpečnosti systému ve tvaru

$$A \text{ -> } \Box A$$

které vyjadřuje fakt, jakmile A platí v nějakém stavu, bude platit i ve všech následujících stavech.

Příklad. (Zakončující čítač (terminating counter)) Γ_{tcount} který je na nějaké množině data(Γ_{tcount}) specifikován temporálními axiomy

(TC1) $c \leq 100$

(TC2) $(on \wedge c < 100) \text{ -> } ((on' \wedge c' = c + 1) \vee (\neg on' \wedge c' = c))$

(TC3) $(\neg on \wedge c < 100) \text{ -> } ((\neg on' \wedge c' = c) \vee (on' \wedge c' = 0))$

(TC4) $c = 100 \text{ -> } (on' \text{ <-> } on \wedge c' = c)$

Zde c a on jsou systémové proměnné pro hodnotu čítače a řízení výpočtu (zapínání, přerušování a vypínání).

Formule

$$c \leq 100 \quad c < 100 \quad c' = c + 1 \quad c' = c \quad \text{ale také} \quad on \quad \neg on$$

jsou výrokové konstanty.

Potom lze odvodit, že pro zakončující čítač platí

$$c = 100 \rightarrow \Box(c = 100)$$

Dá se ukázat, že takto specifikovaný zakončující čítač má i tuto vlastnost

$$c < 100 \rightarrow \Box(c = 100 \rightarrow on)$$

„je-li v nějakém stavu $c < 100$ a v některém z následujících stavů c dosáhne hodnoty 100 potom čítač bude zapnut“.

b) Precedenční vlastnosti

Stejně jako v případě bezpečnostních vlastností se precedenční vlastnosti specifikované formulemi

$$\boldsymbol{A} \rightarrow (\boldsymbol{B} \ ☻ \ \boldsymbol{C})$$

odvodí z indukčních pravidel pro operátory ☻ .

(indunless) $\boldsymbol{A} \rightarrow (\circ \boldsymbol{C} \vee \circ(\mathbf{A} \wedge \mathbf{B})) \mid\!- \ \boldsymbol{A} \rightarrow (\boldsymbol{B} \textbf{ unless } \boldsymbol{C})$

(indunl) $\boldsymbol{A} \rightarrow (\boldsymbol{C} \vee (\mathbf{B} \wedge \circ\mathbf{A})) \mid\!- \ \boldsymbol{A} \rightarrow (\boldsymbol{B} \textbf{ unl } \boldsymbol{C})$

(indatnext) $\boldsymbol{A} \rightarrow (\circ(\boldsymbol{C} \rightarrow \mathbf{B}) \wedge \circ(\neg\mathbf{C} \rightarrow \boldsymbol{A})) \mid\!- \ \boldsymbol{A} \rightarrow (\boldsymbol{B} \textbf{ atnext } \boldsymbol{C})$

(indbefore) $\boldsymbol{A} \rightarrow (\circ \neg \boldsymbol{C} \wedge \circ(\boldsymbol{A} \vee \mathbf{B})) \mid\!- \ \boldsymbol{A} \rightarrow (\boldsymbol{B} \textbf{ before } \boldsymbol{C})$

Ve všech čtyřech případech hraje formule ***A*** podobnou roli jako v pravidle (ind), „je nositelem indukce“. Pokud ***A*** platí v nějakém stavu stavu, pak platí –nyní za určitých podmínek- v dalším stavu.

Příklad. (Ne-zakončující čítač (non-terminating counter)) Γ_{count} který je na nějaké množině data(Γ_{count}) specifikován temporálními axiomy

$$on \;\; \text{->} \; ((on' \wedge \; c' = c + 1) \vee (\neg on' \wedge \; c' = c))$$
$$\neg on \;\; \text{->} \; ((\neg on' \wedge \; c' = c) \vee (on' \wedge \; c' = 0))$$

má vlastnost

$$\neg on \;\; \text{->} \; (c = 0 \;\; \textbf{atnext} \;\; on)$$

ktrerá říká, že vypnutý čítač bude nastaven na hodnotu 0 pokud bude zapnut v příštím stavu.

Pro ne-zakončující čítač se dá odvodit i další vlastnost

$$c = x \text{ -> } (c \geq x \ \textbf{unl}\ c = 0)$$

jakákoliv hodnota čítače nebude v dalších krocích snížena kromě případu, že čítač bude resetován na 0 .

c) „Koncové“ vlastnosti

jsou specifikovány formulemi tvaru

$$A \text{ -> } \Diamond B$$

Pro odvozování takových formulí poskytuje výroková temporální logika (LTL) následující dvě pravidla

(som) $\boldsymbol{A} \rightarrow \circ \boldsymbol{B} \vdash \boldsymbol{A} \rightarrow \Diamond \boldsymbol{B}$

(chain) $\boldsymbol{A} \rightarrow \Diamond \boldsymbol{B}, \boldsymbol{B} \rightarrow \Diamond \boldsymbol{C} \vdash \boldsymbol{A} \rightarrow \Diamond \boldsymbol{C}$

která se dají použít ve velmi jednoduchých důkazech opírajících se o konečné řetězce (posloupnosti) implikací: abychom dokázali

$$\boldsymbol{A} \rightarrow \Diamond \boldsymbol{B}$$

pomocí pravidla (chain), musíme dokázat musíme dokázat dva nebo více „malých kroků“ stejného druhu. Pravidlo (som) je jednoduchý prostředek pro pro ospravedlnění takových „malých kroků“.

I když je tento postup dost často použitelný, pravidla (som) a (chain) jsou slabá. Ovšem v temporální predikátové logice je možné obě pravidla nahradit jedním silnějším, univerzálně použitelným pravidlem. Nebudeme ho uvádět, ale tento fakt vezmeme na vědomí.

Příklad. („Omezený čítač” Γ_{bcount})

Specifikace takového systému v LTL vznikne upravením specifikace zakončujícího čítače následujícím způsobem: při dosažení hodnoty 100, čítač nezakončí výpočet, ale je vypnut Navíc je ve výpočtu povoleno vykonat za sebou až N „prázdných“ kroků, kde N je pevně dané přirozené číslo.

Nebudeme uvádět přesnou formalizaci takového systému jako Stavového přechodového systému, ale napíšeme temporální specifikaci $\mathbf{\Gamma}_{bcount}$:

$c \leq 100 \wedge b \leq N$

$(on \wedge c < 100) \rightarrow ((on' \wedge c' = c + 1) \vee (\neg on' \wedge c' = c \wedge b' = 0))$

$(on \wedge c = 100) \rightarrow (\neg on' \wedge c' = c \wedge b' = 0)$

$(\neg on \wedge b < N) \rightarrow ((\neg on' \wedge c' = c \wedge b' = b + 1) \vee (on' \wedge c' = 0))$

$(\neg on \wedge b = N) \rightarrow (on' \wedge c' = 0))$

Zde b je nová systémová proměnná, která počítá počet prázných kroků. Povšimněme si, že pro jednoduchost specifikace nevylučujeme možnost, že na samém začátku, pokud $b > 0$ systém bude zapnut až po méně něž N prázdných krocích.

Z této specifikace je možné odvodit, že systém $\mathbf{\Gamma}_{bcount}$ má koncovou vlastnost

$$\Diamond(c = 0)$$

vyjadřující fakt, že stav, kdy čítač má hodnotu 0 bude dosažen z kteréhokoli jiného stavu.

Podobně se dá ukázat, že

$$(c > 0 \wedge on) \text{ -> } \Diamond(c > 0 \wedge \neg on)$$

pokud čítač má nenulovou hodnotu, bude někdy později se stejnou hodnotou vypnut .

Ukázali jsme, jak se odvozují tři jednoduché vlastnosti různých systémů z jejich specifikace. Zde neuvedené důkazy nejsou úplně krátké, ale provádějí se ve výrokové temporální logice.

Děkuji vám za přízeň po celý semestr

a za pozornost.