

Temporální logika

1 / 8

VZ 2009



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

V intuitivním smyslu stavové systémy zahrnují „stavy“ a vykazují „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.

Doporučení pro další čtení: Při cvičení se probírají jen části se žlutým podkladem (jako tento odstavec). Ostatní text jen dokresluje sdělovanou informaci.

2 / 8

VZ 2009



Podobně jako v klasické logice je vhodné začít s výrokovou verzí temporální logiky.

Stavy. K popisu stavů použijeme prvotní formule, které vyjadřují jednoduchá tvrzení například „globální proměnná x má hodnotu 10“.

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

K popisu konkrétních systémů obvykle postačí konečná množina prvotních formulí.

3 / 8

VZ 2009



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 očíslováme přirozenými čísly

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

4 / 8

VZ 2009



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 větvcím se čase.

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

- všech formulí z množiny V a
- symbolů **false** \rightarrow \square \circ $()$

5 / 8

VZ 2009



Temporální operátory \circ , \square , \diamond se (anglicky) nazývají

- \circ *nexttime* nebo jen *next*,
- \square *always* nebo *henceforth* a
- \diamond *sometime*.

Formule $\circ A$, $\square A$ a $\diamond A$ se (anglicky) čtou

- $\circ A$: *nextA*, česky *příště A*,
- $\square A$: *alwaysA*, česky *vždy A*
- $\diamond A$: *sometimeA*, česky *někdy A*

Preference \neg , \circ , \square , \diamond váží silněji než \vee , \wedge , \rightarrow a \equiv má nejslabší prioritu.

6 / 8

VZ 2009



Temporální operátory a přirozený jazyk

- α nechť znamená „Měsíc obíhá Zemi“,
- β nechť znamená „Měsíc vychází“,
- γ nechť znamená „Měsíc zapadá“.

- $\diamond\beta$, která vyjadřuje tvrzení „Měsíc někdy vyjde“,
- $\square\diamond\beta$ s významem „Měsíc bude vycházet znovu a znovu“,
- $\square(\beta \rightarrow \diamond\gamma)$ s významem „vždy po východu měsíce bude někdy následovat jeho západ“.

7/8

VZ 2009



Definice. (Formule LTL)

1. Každá prvotní formule z množiny V je formule.
2. Výroková konstanta **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 $\square A$ jsou formule.

Ostatní spojky se zavádějí jako zkratky

- $\neg A$ je zkratka za formuli $\neg(\mathbf{false} \rightarrow A)$
- **true** je zkratka za formuli $\neg \mathbf{false}$
- \vee, \wedge, \equiv jako v klasické logice,
- $\diamond A$ je zkratka za formuli $\neg \square \neg A$

8/8

VZ 2009



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

- $A \rightarrow \circ B$ „jestliže A potom B v dalším stavu“
- $A \rightarrow \square B$ „jestliže A potom B teď a v každém dalším stavu“
- $A \rightarrow \diamond B$ „jestliže A potom B teď nebo v nějakém dalším stavu“
- $\square(A \rightarrow B)$ „jestliže A teď nebo v nějakém dalším stavu potom B platí ve stejném stavu“
- $\square \diamond A$ „teď a za každým dalším stavem někdy platí A “
„(od teď) A bude platit v nekonečně mnoha stavech“
„formule A bude nastavát znovu a znovu“
- $\diamond \square A$ „od některého budoucího stavu bude stále platit A “
„ A platí skoro vždycky (od jistého okamžiku dál)“

9 / 8

VZ 2009



Popis komplexního systému – dobře fungující úřad

- Jeli požadavek podán, pak bude někdy doručen na správné místo $\square(\text{požadavek} \rightarrow \diamond \text{doručen})$,
- Je-li požadavek doručen, pak jeho zpracování bude zahájeno hned v následujícím okamžiku $\square(\text{doručen} \rightarrow \circ \text{zpracováván})$,
- Zpracováváný požadavek bude jednou vyřízen (hotov) a pak už zůstane hotov (nebude se znovu otevírat) $\square(\text{zpracováván} \rightarrow \diamond \square \text{hotov})$.

Uvedené formule charakterizují chování komplexního systému, označíme je jako program P pro tento systém. Zdá se, že nemůže nastat situace, ve které by systém stále vysílal stejný požadavek, ale ten by nebyl nikdy hotov. Lze se o tom přesvědčit? Na přednášce ukážeme, že formule $\square \text{požadavek} \ \& \ \square \neg \text{hotov}$ nemůže platit, čili tato formule je s uvedeným programem P nekonzistentní.

10 / 8

VZ 2009



V klasické logice se pravdivostní hodnoty výrokových formulí určují z pravdivostního ohodnocení prvotních výroků (tedy v jediné interpretaci, v jednom 'stavu').

V temporální logice, kde se výrokové formule tvoří také z prvotních formulí za použití modálních operátorů \Box , \Diamond a \circ se pravdivostní hodnoty musí určovat 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 prvotních formulí.

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

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

... které nazýváme *stavy*.

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

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

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

Pro temporální strukturu M , index i a formuli A *induktivně* definujeme **pravdivostní hodnotu** $M_i(A)$, neformálně, pravdivostní hodnotu formule A v časovém bodě m_i .

$$M_i(v) = \eta_i(v) \quad v \in V$$

$$M_i(\text{false}) = \text{ff}$$

$$M_i(A \rightarrow B) = \text{tt} \text{ právě když}$$

$$M_i(A) = \text{ff} \text{ nebo } M_i(B) = \text{tt}$$

13 / 8

VZ 2009



Pro operátory

$$M_i(\circ A) = M_{i+1}(A)$$

$$M_i(\Box A) = \text{tt} \text{ právě když } M_j(A) = \text{tt} \text{ pro každé } j \geq i$$

Pro definované symboly

$$M_i(\neg A) = \text{tt} \text{ právě když } M_i(A) = \text{ff}$$

$$M_i(A \vee B) = \text{tt} \text{ právě když}$$

$$M_i(A) = \text{tt} \text{ nebo } M_i(B) = \text{tt}$$

$$M_i(A \wedge B) = \text{tt} \text{ právě když}$$

$$M_i(A) = \text{tt} \text{ a } M_i(B) = \text{tt}$$

14 / 8

VZ 2009



$M_i(A \equiv B) = \text{tt}$ právě když $M_i(A) = M_i(B)$

$M_i(\text{true}) = \text{tt}$

$M_i(\Diamond A) = \text{tt}$ právě když $M_j(A) = \text{tt}$ 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

$M_i(\Diamond A) = \text{tt}$ právě když $M_i(\neg \Box \neg A) = \text{tt}$

právě když $M_i(\Box \neg A) = \text{ff}$

právě když $M_j(\neg A) = \text{ff}$ pro nějaké $j \geq i$

právě když $M_j(A) = \text{tt}$ pro nějaké $j \geq i$

15 / 8

VZ 2009



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 M pro V , (nebo že A je splněna v M) a píšeme $M \models A$ nebo $\models_M A$, jestliže $M_i(A) = \text{tt}$ pro všechna i

Říkáme, že struktura M je modelem množiny formulí T , jestliže $M \models B$ 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 M množiny formulí T .

16 / 8

VZ 2009



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

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

Je třeba ukázat, že $M_i(\neg \circ A) = M_i(\circ \neg A)$ platí pro každou strukturu M a všechny časové body i .

$$\begin{aligned} M_i(\neg \circ A) = tt &\Leftrightarrow M_i(\circ A) = ff \\ &\Leftrightarrow M_{i+1}(A) = ff \\ &\Leftrightarrow M_{i+1}(\neg A) = tt \\ &\Leftrightarrow M_i(\circ \neg A) = tt \end{aligned}$$

17 / 8

VZ 2009



Lemma 1. (korektnost pravidla modus ponens)

Nechť M je temporální struktura a $i \in \mathbb{N}$ (množina přirozených čísel), nechť $M_i(A) = tt$ a $M_i(A \rightarrow B) = tt$, potom $M_i(B) = tt$.

Důkaz. $M_i(A \rightarrow B) = tt$, tedy $M_i(A) = ff$ nebo $M_i(B) = tt$.

Z předpokladu $M_i(A) = tt$ dostáváme $M_i(B) = tt$.

Věta 2. Je-li $\mathbf{T} \models A$ a $\mathbf{T} \models A \rightarrow B$, potom $\mathbf{T} \models B$.

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

$$M_i(A) = M_i(A \rightarrow B) = tt$$

a podle lemmatu 1 pak také $M_i(B) = tt$. Tedy $\mathbf{T} \models B$.

18 / 8

VZ 2009



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

Důkaz. Necht' \mathbf{M} je libovolná temporální struktura, která je modelem \mathbf{T} a necht' i je přirozené číslo.

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

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

To znamená, že

$$\mathbf{T} \not\models \circ A \quad \text{a} \quad \mathbf{T} \not\models \Box A.$$

19 / 8

VZ 2009



Věta 4. Je-li $\mathbf{T} \models A \rightarrow B$ a $\mathbf{T} \models A \rightarrow \circ A$, potom
 $\mathbf{T} \models A \rightarrow \Box B$ (pravidlo indukce).

Důkaz. Necht' \mathbf{M} je struktura pro \mathbf{T} . Rozlišíme 2 možnosti:

- Je-li $\mathbf{M} \models \neg A$, je tvrzení věty důsledkem vlastností spojky „ \rightarrow “.
- Neplatí-li $\mathbf{M} \models \neg A$, zvolme i nejmenší takové, že $\mathbf{M}_i \models A$. Tvrzení $\mathbf{M}_i \models \Box A$, $\mathbf{M}_i \models \Box B$ jsou jednoduchými důsledky předpokladů věty. Tedy pro všechna j platí $\mathbf{M}_j \models A \rightarrow \Box B$.

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

$$\mathbf{M}^i = (\eta_0', \eta_1', \eta_2', \dots)$$

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

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

20 / 8

VZ 2009



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

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

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

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

Obrácená implikace v LTL neplatí! Stačí uvažovat případ, kdy \mathbf{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 \mathbf{M} , kde pro nějaké i a $j > i$ platí $\mathbf{M}_i(A) = \text{tt}$ a $\mathbf{M}_j(A) = \text{ff}$. Stačí uvažovat případ, kdy A je některá výroková konstanta.

21 / 8

VZ 2009



Binární temporální operátory

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

Definice. $A \cup B$

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

$$\mathbf{M}_i(A \cup B) = \text{tt} \Leftrightarrow \begin{array}{l} \mathbf{M}_j(B) = \text{tt} \text{ pro nějaké } j, j > i \text{ a} \\ \mathbf{M}_k(A) = \text{tt} \text{ pro každé } k, i < k < j \end{array}$$

22 / 8

VZ 2009



Důležité validní formule

$$\begin{aligned} & \models \Box \neg A \equiv \neg \Diamond A & \models \Diamond \neg A \equiv \neg \Box A \\ (1) \quad & \models \Box \neg A \equiv \neg \Box A & \\ & \models \Box \Diamond \neg A \equiv \neg \Diamond \Box A & \models \Diamond \Box \neg A \equiv \neg \Box \Diamond A \\ & \not\models ((\neg A) \cup B) \equiv \neg (A \cap B) \end{aligned}$$

Následující validní implikace nelze zesílit na ekvivalence

$$\begin{aligned} & \models A \rightarrow \Diamond A & \models \Box A \rightarrow A \\ & \models \Box A \rightarrow \Diamond A & \models \Box A \rightarrow \Box A \\ & \models \Box A \rightarrow \Diamond \Box A & \models \Box A \rightarrow \Box \Box A \\ & \models A \cup B \rightarrow \Diamond A & \models \Diamond \Box A \rightarrow \Box \Diamond A \end{aligned}$$

23 / 8

VZ 2009

**Idempotence** $\Diamond, \Box, \Box\Diamond$ a $\Diamond\Box$

$$\begin{aligned} & \models \Diamond\Diamond A \equiv \Diamond A & \models \Box\Box A \equiv \Box A \\ & \models \Diamond\Box\Diamond A \equiv \Diamond\Box A & \not\models \Box\Diamond\Box\Diamond A \equiv \Box\Diamond A \end{aligned}$$

Ale operátor příštího stavu není idempotentní. Formule $\Box\Box A \equiv \Box A$ není validní.

Vlastnosti nekonečných modalit $\Box\Diamond$ a $\Diamond\Box$: „konzumují“ všechny ostatní modalities s jedním argumentem, které jsou na ně aplikované. S menším násilím na syntax formulí to můžeme kompaktně vyjádřit takto

$$\begin{aligned} & \models (\Box\Diamond) A \equiv \Diamond (\Box\Diamond) A \equiv \Diamond (\Box\Diamond) A \equiv \Box (\Box\Diamond) A \\ & \models \Box\Diamond (\Box\Diamond) A \equiv \Diamond\Box (\Box\Diamond) A \\ & \models (\Diamond\Box) A \equiv \Box (\Diamond\Box) A \equiv \Box (\Diamond\Box) A \equiv \Diamond (\Diamond\Box) A \\ & \models \Box\Diamond (\Diamond\Box) A \equiv \Diamond\Box (\Diamond\Box) A \end{aligned}$$

24 / 8

VZ 2009



Podle své definice jsou operátory \diamond a \Box povahy *existenční* a operátory \square a \Diamond 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.

$$\begin{aligned} \models \diamond(A \vee B) &\equiv (\diamond A \vee \diamond B) & \models \square \diamond(A \vee B) &\equiv (\square \diamond A \vee \square \diamond B) \\ \models \square(A \wedge B) &\equiv (\square A \wedge \square B) & \models \diamond \square(A \wedge B) &\equiv (\diamond \square A \wedge \diamond \square B) \end{aligned}$$

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

$$\begin{aligned} \models ((A \wedge B) \cup C) &\equiv ((A \cup C) \wedge (B \cup C)) \\ \models (A \cup (B \vee C)) &\equiv ((A \cup C) \vee (B \cup C)) \end{aligned}$$

25 / 8

VZ 2009



Operátor \circ (*next*) se vztahuje k jedinému časovému bodu, proto se distribuuje se všemi boolovskými spojky.

$$\begin{aligned} \models \circ(A \vee B) &\equiv (\circ A \vee \circ B) & \models \circ(A \wedge B) &\equiv (\circ A \wedge \circ B) \\ \models \circ(A \rightarrow B) &\equiv (\circ A \rightarrow \circ B) & \models \circ(A \equiv B) &\equiv (\circ A \equiv \circ B) \end{aligned}$$

přitom ekvivalence $\models \circ \neg A \equiv \neg \circ A$ již byla uvedena výše.

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

$$\begin{aligned} \models (\square A \vee \square B) &\rightarrow \square(A \vee B) & \models \diamond \square(A \vee B) &\rightarrow (\diamond \square A \vee \diamond \square B) \\ \models \diamond(A \wedge B) &\rightarrow (\diamond A \wedge \diamond B) & \models \square \diamond(A \wedge B) &\rightarrow (\square \diamond A \wedge \square \diamond B) \end{aligned}$$

$$\begin{aligned} \models ((A \cup C) \vee (B \cup C)) &\rightarrow ((A \vee B) \cup C). \\ \models (A \cup (B \wedge C)) &\rightarrow ((A \cup B) \wedge (A \cup C)) \end{aligned}$$

26 / 8

VZ 2009



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

$$\begin{aligned} & \models \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B) & \models \Box(A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B) \\ & \models \circ(A \rightarrow B) \rightarrow (\circ A \rightarrow \circ B) \\ & \models \Box(A \rightarrow B) \rightarrow (\Diamond \Box A \rightarrow \Diamond \Box B) & \models \Box(A \rightarrow B) \rightarrow (\Diamond \Box A \rightarrow \Diamond \Box B) \\ & \models \Box(A \rightarrow B) \rightarrow ((A \cup C) \rightarrow (B \cup C)) \\ & \models \Box(A \rightarrow B) \rightarrow ((C \cup A) \rightarrow (C \cup B)) \\ & \models (A \cup B) \equiv B \vee (A \wedge \circ(A \cup B)) \end{aligned}$$

Nakonec uvedeme důležité charakteristiky temporálních operátorů pomocí pevných bodů.

$$\models \Diamond A \equiv (A \vee \circ \Diamond A) \quad (3) \quad \models \Box A \equiv (A \wedge \circ \Box A)$$

Pozor : sice platí $\models \Box A \rightarrow (A \rightarrow \circ \Box A)$, ale formule popisující opačnou implikaci $(A \rightarrow \circ \Box A) \rightarrow \Box A$ určitě neplatí !

27 / 8

VZ 2009



Příklady zdrojů dalších informací:

- Huth M., Ryan M.: *Logic in Computer Science*, Cambridge University Press, 2004
- Programový systém [SPIN](#)

28 / 8

VZ 2009

