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 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.

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

Ingredience

Stavy. K popisu stavů použijeme prvotní formule, které vyjadřují jednoduchá tvrzeni 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.

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 prvotních formulí, jazyk výrokové temporální logiky L_{LTL} sestává ze

- všech formulí z množiny V a
- symbolů **false** $\rightarrow$ $\Box$ $\bigcirc$ ( )

Temporální operátory $\bigcirc$, $\Box$, $\Diamond$ se (anglicky) nazývají

$\bigcirc$ *nexttime* nebo jen *next* ,

$\Box$ *always* nebo *henceforth* a

$\Diamond$ *sometime.*

Formule $\bigcirc A$, $\Box A$ **a** $\Diamond A$ se (anglicky) čtou

$\bigcirc A$: *nextA* , česky *příště A,*

$\Box A$: *alwaysA*, česky *vždy A*

$\Diamond A$: *sometimeA*, česky *někdy A*

Preference $\neg$, $\bigcirc$, $\Box$, $\Diamond$ váží silněji než $\vee$, $\wedge$, $\rightarrow$ a $\equiv$ má nejslabší prioritu.

Definice. (Formule LTL)

1. Každá prvotní formule 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 zavádějí jako zkratky

- $\neg A$ je zkratka za formuli $\neg$ (**false** $\rightarrow A$)
- **true** je zkratka za formuli $\neg$ **false**
- $\vee$, $\wedge$, $\equiv$ jako v klasické logice ,
- $\Diamond A$ je zkratka za formuli $\neg \Box \neg A$

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, 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 prvotních formulí.

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

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

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 prvotních formulí 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 .

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

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

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

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

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

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

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

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

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

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

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

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

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 $\mathbf{K}_i(A) = \text{tt}$ *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. $\neg\circ A \equiv \circ\neg A$ 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.

$\mathbf{K}_i(\neg\circ A) = \text{tt} \Leftrightarrow \mathbf{K}_i(\circ A) = \text{ff}$

$\Leftrightarrow \mathbf{K}_{i+1}(A) = \text{ff}$

$\Leftrightarrow \mathbf{K}_{i+1}(\neg A) = \text{tt}$

$\Leftrightarrow \mathbf{K}_i(\circ\neg A) = \text{tt}$

Lemma 1. (korektnost pravidla modus ponens)

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

Důkaz. $\mathbf{K}_i(A \to B) = \text{tt}$, 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 $T \models A \to B$ potom $T \models B$.

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

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

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 $\mathbf{K}$ je libovolná temporální struktura, která je modelem T a nechť i je přirozené číslo.

Podle předpokladu platí $K_j(A)$ pro všechna j, tedy také

$K_{i+1}(A) = \text{tt}$ a $K_j(A) = \text{tt}$ pro všechna j, $j \geq i$.

To znamená, že

$T \models \circ A$ a $T \models \Box A$.

Zajímavější je následující tvrzení, které uvádíme bez důkazu.

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

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

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

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

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

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

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

Věta 7. Je-li $T \models A \to B$ *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 $A \to \Box A$ nemusí být validní formule. Není pravdivá v žádné temporální struktuře K, kde pro nějaké i a $j > i$ platí $K_i(A) = \text{tt}$ a $K_j(A) = \text{ff}$. Stačí uvažovat případ, kdy A je některá výroková konstanta.

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

$A \to \circ B$ „jestliže A potom B v dalším stavu“

$A \to \Box B$ „jestliže A potom B teď a v každém dalším stavu“

$A \to \Diamond B$ „jestliže A potom B teď nebo v nějakém dalším stavu“

$\Box(A \to B)$ „jestliže A teď nebo v nějakém dalším stavu potom B platí ve stejném stavu“

$\Box\circ A$ „teď a za každým dalším stavem někdy platí A“

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

$\circ\Box A$ „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*“ .

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

jinak vyjádřeno

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

Důležité validní formule

(1)
$\models \Box\neg A \equiv \neg\Diamond A$ $\quad \models \Diamond\neg A \equiv \neg\Box A$

$\models \circ\neg A \equiv \neg\circ A$

$\models \Box\Diamond\neg A \equiv \neg\Diamond\Box A$ $\quad \models \Diamond\Box\neg A \equiv \neg\Box\Diamond A$

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

Následující validní implikace nelze zesílit na ekvivalence

$\models A \to \Diamond A$ $\quad \models \Box A \to A$

$\models \circ A \to \Diamond A$ $\quad \models \Box A \to \circ A$

$\models \Box A \to \Diamond A$ $\quad \models \Box A \to \circ\Box A$

$\models A \cup B \to \Diamond A$ $\quad \models \Diamond\Box A \to \Box\Diamond A$

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

$\models \Diamond\Diamond A \equiv \Diamond A$ $\quad \models \Box\Box A \equiv \Box A$

$\models \Diamond\Box\Diamond\Box A \equiv \Diamond\Box A$ $\quad \models \Box\Diamond\Box\Diamond A \equiv \Box\Diamond A$

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

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

$\models (\Box\Diamond) A \equiv \circ(\Box\Diamond) A \equiv \Diamond(\Box\Diamond) A \equiv \Box(\Box\Diamond) A$

$\models \Box\Diamond(\Box\Diamond) A \equiv \Diamond\Box(\Box\Diamond) A$

$\models (\Diamond\Box) A \equiv \circ(\Diamond\Box) A \equiv \Diamond(\Diamond\Box) A \equiv \Box(\Diamond\Box) A$

$\models \Box\Diamond(\Diamond\Box) A \equiv \Diamond\Box(\Diamond\Box) A$

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.

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

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

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

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

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

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

$\models \bigcirc(A \vee B) \equiv (\bigcirc A \vee \bigcirc B)$ $\quad \models \bigcirc(A \wedge B) \equiv (\bigcirc A \wedge \bigcirc B)$
$\models \bigcirc(A \rightarrow B) \equiv (\bigcirc A \rightarrow \bigcirc B)$ $\quad \models \bigcirc(A \equiv B) \equiv (\bigcirc A \equiv \bigcirc B)$

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

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

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

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

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

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

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

Nakonec uvedeme důležité charakteristiky temporálních operátorů pomocí pevných bodů.

$\models \Diamond A \equiv A \vee \bigcirc\Diamond A$ (3) $\quad \models \Box A \equiv A \wedge \bigcirc\Box A$
$\models (A \cup B) \equiv B \vee (A \wedge \bigcirc(A \cup B))$
$\models (A \mathrm{B} B) \equiv \neg B \wedge (A \vee \bigcirc(A \mathrm{B} B))$

Příklady zdrojů dalších informací:

- Huth M., Ryan M.: *Logic in Computer Science*, Cambridge University Press, 2004
- Programový systém SPIN