

Ověřování korektnosti reaktivních systémů

Nechť *Impl* je implementace systému (např. v CCS syntaxi).

Přístup ověřování ekvivalencí (Equivalence Checking)

$$Impl \equiv Spec$$

- \equiv je abstraktní ekvivalence, např. \sim nebo \approx
- *Spec* je často vyjádřena stejným jazykem jako *Impl*
- *Spec* poskytuje plnou specifikaci požadovaného chování

Ověřování korektnosti reaktivních systémů

Nechť *Impl* je implementace systému (např. v CCS syntaxi).

Přístup ověřování ekvivalencí (Equivalence Checking)

$$Impl \equiv Spec$$

- \equiv je abstraktní ekvivalence, např. \sim nebo \approx
- *Spec* je často vyjádřena stejným jazykem jako *Impl*
- *Spec* poskytuje plnou specifikaci požadovaného chování

Přístup ověřování modelů (Model Checking)

$$Impl \models Property$$

- \models je relace splňování (satisfaction relation)
- *Property* je konkrétní vlastnost, často vyjádřená prostřednictvím formulí vhodné logiky
- *Property* je částečná specifikace požadovaného chování

Cíl

Vytvořit logiku, ve které můžeme vyjádřit zajímavé vlastnosti reaktivních systémů.

Modální vlastnosti (Modal Properties) – co se může stát teď (možnost, nutnost)

- pije kávu (může pít kávu právě teď)
- nepije čaj
- pije kávu i čaj
- pije vodu po kávě

Modální vlastnosti (Modal Properties) – co se může stát teď (možnost, nutnost)

- pije kávu (může pít kávu právě teď)
- nepije čaj
- pije kávu i čaj
- pije vodu po kávě

Temporální vlastnosti (Temporal Properties) – chování v čase

- nikdy nepije žádný alkohol
(**vlastnost bezpečnosti** (safety property): nic špatného se nemůže stát)
- nakonec (ve smyslu někdy v budoucnu, dříve nebo později) si dá sklenku vína
(**vlastnost živosti** (liveness property): něco dobrého se stane)

Modální vlastnosti (Modal Properties) – co se může stát teď (možnost, nutnost)

- pije kávu (může pít kávu právě teď)
- nepije čaj
- pije kávu i čaj
- pije vodu po kávě

Temporální vlastnosti (Temporal Properties) – chování v čase

- nikdy nepije žádný alkohol
(**vlastnost bezpečnosti** (safety property): nic špatného se nemůže stát)
- nakonec (ve smyslu někdy v budoucnu, dříve nebo později) si dá sklenku vína
(**vlastnost živosti** (liveness property): něco dobrého se stane)

Můžeme tyto vlastnosti vyjádřit použitím ověřování ekvivalencí?

Syntaxe formulí ($a \in Act$)

$$F, G ::= tt \mid ff \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$$

Syntaxe formulí ($a \in Act$)

$$F, G ::= tt \mid ff \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$$

Intuice:

tt tuto vlastnost má každý proces (neboli každý proces splňuje tuto formuli)

ff tuto vlastnost nemá žádný proces

\wedge, \vee běžná logická konjunkce a disjunkce

$\langle a \rangle F$ existuje alespoň jeden následovník přes akci a , který splňuje F

$[a]F$ každý následovník přes akci a musí splňovat F

Syntaxe formulí ($a \in Act$)

$$F, G ::= tt \mid ff \mid F \wedge G \mid F \vee G \mid \langle a \rangle F \mid [a]F$$

Intuice:

tt tuto vlastnost má každý proces (neboli každý proces splňuje tuto formuli)

ff tuto vlastnost nemá žádný proces

\wedge, \vee běžná logická konjunkce a disjunkce

$\langle a \rangle F$ existuje alespoň jeden následovník přes akci a , který splňuje F

$[a]F$ každý následovník přes akci a musí splňovat F

Poznámka

Tato logika nezahrnuje temporální vlastnosti jako **vždy/nikdy v budoucnu** nebo **nakonec** (eventually).

Nechť $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ je LTS.

Platnost $p \models F$, kde $p \in Proc$, F je HM formule

$p \models tt$ pro každé $p \in Proc$

$p \models ff$ pro žádné p (píšeme tedy $p \not\models ff$)

$p \models F \wedge G$ právě tehdy, když $p \models F$ a $p \models G$

$p \models F \vee G$ právě tehdy, když $p \models F$ nebo $p \models G$

$p \models \langle a \rangle F$ právě tehdy, když $p \xrightarrow{a} p'$ pro nějaké $p' \in Proc$
takové, že $p' \models F$

$p \models [a]F$ právě tehdy, když $p' \models F$ pro každé $p' \in Proc$
takové, že $p \xrightarrow{a} p'$

Píšeme $p \not\models F$ kdykoliv p nespĺňuje F .

A co negace?

Pro každou formuli F definujeme formuli F^c následovně:

- $tt^c = ff$
- $ff^c = tt$
- $(F \wedge G)^c = F^c \vee G^c$
- $(F \vee G)^c = F^c \wedge G^c$
- $(\langle a \rangle F)^c = [a]F^c$
- $([a]F)^c = \langle a \rangle F^c$

A co negace?

Pro každou formuli F definujeme formuli F^c následovně:

- $tt^c = ff$
- $ff^c = tt$
- $(F \wedge G)^c = F^c \vee G^c$
- $(F \vee G)^c = F^c \wedge G^c$
- $(\langle a \rangle F)^c = [a]F^c$
- $([a]F)^c = \langle a \rangle F^c$

Věta (F^c je ekvivalentní negaci F)

Pro každý proces $p \in Proc$ a každou HM formuli F

- 1 $p \models F \implies p \not\models F^c$
- 2 $p \not\models F \implies p \models F^c$

Hennessy-Milner logika – denotační sémantika

Pro formuli F budeme $\llbracket F \rrbracket \subseteq Proc$ označovat množinu všech stavů, které splňují F .

Denotační sémantika: $\llbracket _ \rrbracket : Formule \rightarrow 2^{Proc}$

- $\llbracket tt \rrbracket = Proc$
- $\llbracket ff \rrbracket = \emptyset$
- $\llbracket F \vee G \rrbracket = \llbracket F \rrbracket \cup \llbracket G \rrbracket$
- $\llbracket F \wedge G \rrbracket = \llbracket F \rrbracket \cap \llbracket G \rrbracket$
- $\llbracket \langle a \rangle F \rrbracket = \langle \cdot a \cdot \rangle \llbracket F \rrbracket$
- $\llbracket [a] F \rrbracket = [\cdot a \cdot] \llbracket F \rrbracket$

kde $\langle \cdot a \cdot \rangle, [\cdot a \cdot] : 2^{(Proc)} \rightarrow 2^{(Proc)}$ jsou definovány následovně:

$$\langle \cdot a \cdot \rangle S = \{ p \in Proc \mid \exists p' (p \xrightarrow{a} p' \wedge p' \in S) \}$$

$$[\cdot a \cdot] S = \{ p \in Proc \mid \forall p' (p \xrightarrow{a} p' \Rightarrow p' \in S) \}.$$

Věta

Nechť $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ je LTS, $p \in Proc$ a F je formule Hennessy-Milner logiky. Potom

$$p \models F \quad \text{právě tehdy, když} \quad p \in \llbracket F \rrbracket.$$

Důkaz: indukcí podle struktury formule F .

Image-Finite systém

Nechť $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ je LTS. Nazveme jej **image-finite** právě tehdy, když pro každé $p \in Proc$ a každé $a \in Act$ je množina

$$\{p' \in Proc \mid p \xrightarrow{a} p'\}$$

konečná.

Věta (Hennessy-Milner)

Nechť $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ je image-finite LTS a $p, q \in Proc$.
Potom

$$p \sim q$$

právě tehdy, když

pro každou HM formuli F : $(p \models F \iff q \models F)$.