SAV – opakovací příklady – 13. 12. 2023

1. Analýza programu pomocí predikátové abstrakce a "lazy abstraction", a to s $\mathcal{P}_1 = \emptyset$ (počáteční mn-a predikátů).

	1. běh dopředný s $\mathcal{P}_1=\emptyset$	2. běh – zpětný		důkaz míče: 3. běh dopředný s $\mathcal{P}_2$
	$\top$	φ		
`t:=0;`	$\top$	$\exists x''.\ x''=0 \wedge (\exists y'.\ y'=x''-1 \wedge (\exists x'(\exists t'(t'=t+1 \wedge \exists z'(z'=x'+1 \wedge y'>z')) \wedge x'>0)))$	SAT? ×	$\top$
`x:=0;`	$\top$	$\exists y'.\ y'=x-1 \wedge (\exists x'(\exists t'(t'=t+1 \wedge \exists z'(z'=x'+1 \wedge y'>z')) \wedge x'>0))$	SAT? ✓ (např.	$x=0 \wedge \neg(x>0) \wedge \neg(x<-2)$
`y:=x-1;`	$\top$	$(\exists x'(\exists t'(t'=t+1 \wedge \exists z'(z'=x'+1 \wedge y>z')) \wedge x'>0))$	SAT? ✓ $t=0, t'=1, x'=1, x=4, \ldots$)	$x=0 \wedge \neg(x>0) \wedge \neg(x<-2) \wedge y=x-1 \wedge y=-1 \wedge \neg(y>x+1)$; $y=-1$
≡? `read(&x);`	$\top$	$(\exists t'(t'=t+1 \wedge \exists z'(z'=x+1 \wedge y>z')) \wedge x>0)$	SAT? ✓ (*)	
`if (x>0){`	$\top$	$\exists t'(t'=t+1 \wedge \exists z'(z'=x+1 \wedge y>z'))$	SAT? ✓	$\varphi \{ y=-1 \wedge \neg(y=x-1) \wedge \neg(x=0) \wedge x>0 \wedge \neg(y>x+1) \wedge \neg(x<-2)$
`t:=t+1;`	$\top$	$\exists z'(z'=x+1 \wedge y>z')$	SAT? ✓	— " —
`z:=x+1;`	$\top$	$\top \wedge (y>z)$	SAT? ✓	$\neg(y>z) \wedge y=-1 \wedge \neg(y=x-1) \wedge \neg(x=0) \wedge x>0 \wedge \neg(y>x+1) \wedge \neg(x<-2) \wedge z=x+1$
assert(y≤z) `if (y>z) error();`	$\top$	$\top$		$\models$

`}` ⋮

reachability reagion (RR)

bad region (BR)

u SAT dotazu jsou volné pr. implicitně exist. kvantifikovány.

Existuje budoucí hodnota z taková, že… → nějak se spočte a něco v budoucnu platí

* z toho plyne, že $y>x+1 \wedge x>0$ – lze splnit, např. $x=1, y=3$.

- důkaz nesplnitelnosti φ:
  - $x''=0 \wedge y'=x''-1 \Rightarrow y'=-1$
  - $z'=x'+1 \wedge y'>z' \Rightarrow y'>x'+1$
  - $y'=-1 \wedge y'>x'+1 \Rightarrow -1>x'+1 \equiv x'<-2$
  - $x'<-2 \wedge x'>0 \Rightarrow false$
- následně zjemnění vezme jako nové predikáty všechny atomy z důkazu, které nejsou v P_1 (to je aktuálně $\emptyset$)
  - z atomů a a $\neg a$ stačí vzít jen a (a jejich negaci), bez apostrofů a jen splnitelné
  - dostaneme $P_2 = \{ x=0,\ y=x-1,\ y=-1,\ z=x+1,\ x>0,$
    $y>z,\ y>x+1,\ x<-2 \}$
- ilustrace výpočtu, zda po provedení $z:=x+1$ ze stavů odpovídajících ψ bude/nebude platit $y>z$.

  a) - $wp(y>z,\ z:=x+1) = (y>x+1)$
  - $\psi \overset{?}{\Rightarrow} y>x+1$

    $(\dots \wedge \neg(y>x+1) \wedge \dots) \overset{?}{\Rightarrow} y>x+1$ – v tomto případě stačí ověřit SAT levé strany – ta je SAT: napři: $y=-1,\ x=1$ je model a tedy $\Rightarrow$ neplatí !!!

b) – $wp(\neg(y>z),\ z:=x+1) = \neg(y>x+1)$

– $(\ldots \wedge \neg(y>x+1) \wedge \ldots) \overset{?}{\Rightarrow} \neg(y>x+1)$ – PLATÍ!

2. Výpočet nových predikátů pro předchozí příklad pomocí Craigových interpolantů

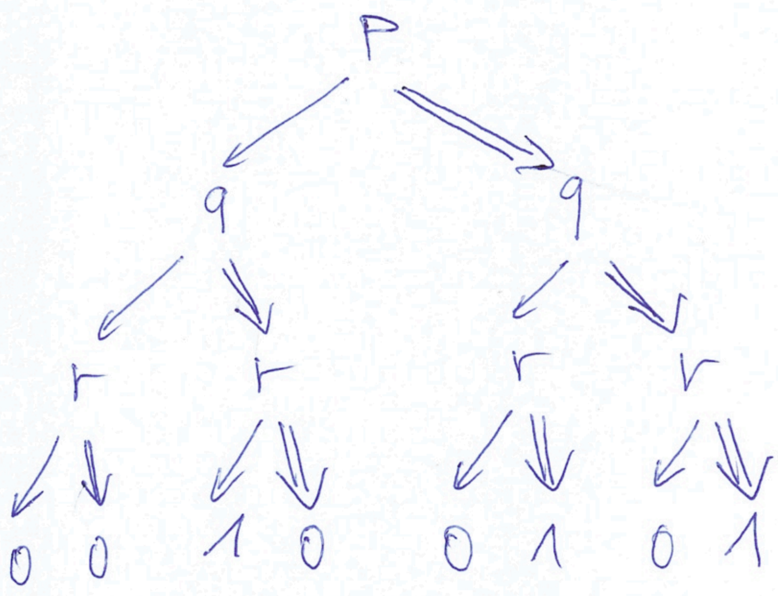
formule nad cestou	Craigovy interpolanty pro příslušné řádky	následoval by dopředný běh s příslušnými predikáty z cr. int.
	T	T – pouze na odpovídajících řádcích
$t_1 = 0 \wedge$	T (sdíl. prom.: t_1, šlo by $t_1=0$ – ale není třeba)	T
$x_1 = 0 \wedge$	$x_1=0$ (sdíl. prom.: t_1, x_1)	$x=0$
$y_1 = x_1 - 1 \wedge$	$y_1=-1$ (sdíl. prom.: t_1, y_1)	$y=-1$
read = true $\wedge$	$y_1=-1$	$y=-1$
$x_2 > 0 \wedge$	$y_1=-1 \wedge x_2>0$ (sdíl. prom.: y_1, x_2, t_1)	$y=-1 \wedge x>0$
$t_2 = t_1 + 1 \wedge$	$y_1=-1 \wedge x_2>0$ (sdíl. prom. y_1, x_2)	$y=-1 \wedge x>0$
$z_1 = x_2 + 1 \wedge$	$y_1=-1 \wedge z_1>1$ (sdíl. prom. y_1, z_1) – lze užít také obecnější $\neg(y_1>z_1)$	$y=-1 \wedge z>1$
$y_1 > z_1$	F	F

read – žádné omezení, ale nová kopie x

UNSAT

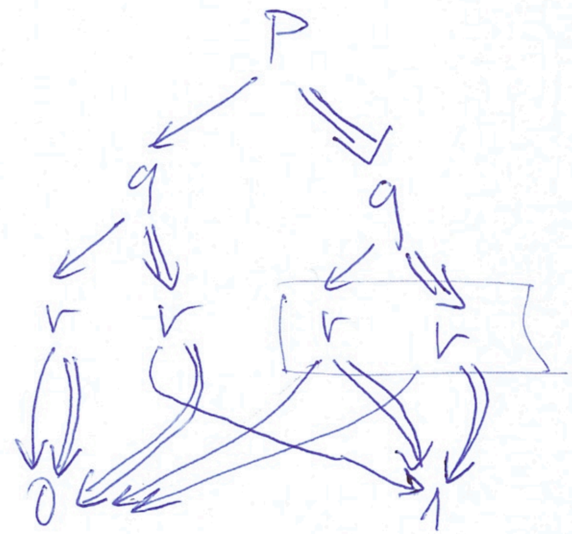
$(x_2>0 \wedge z_1>1 \wedge y_1=-1 \wedge y_1>z_1)$

3. Zkonstruujte rozhodovací strom pro formuli
$(\neg p \Rightarrow (q \wedge \neg r)) \wedge (p \Rightarrow r)$ pro
usp. proměnných $p < q < r$ a ten následně
algoritmicky převeďte na RO BDD.

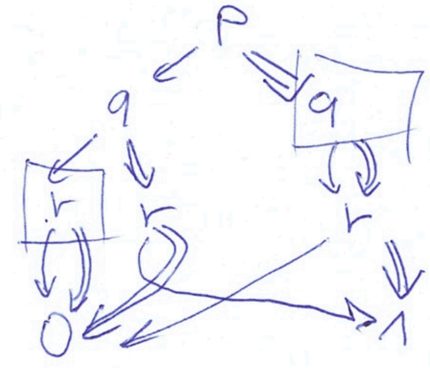


rozhodovací strom
(neredukované OBDD)

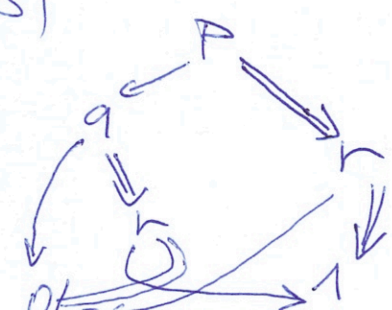
⇝ odstranění redundantních listů



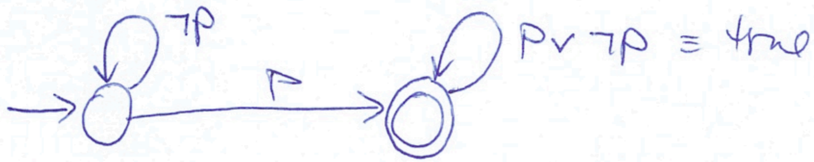
⇝ sloučení duplicitních uzlů

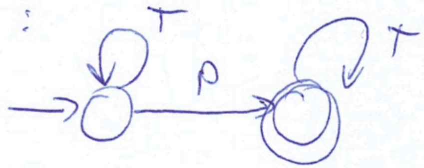


2x
odstranění zbytečných uzlů



4. Převeďte výše uvedenou LTL formuli na BA
(a) ručně / intuitivně , (b) algoritmicky : $\varphi \equiv Fp \equiv \Diamond p$

ad a) 

/ Lze : 

ad b) 1. převedeme na použití zákl. spojek :

$$\Diamond p \equiv true \, U \, p \equiv (p \vee \neg p) \, U \, p \equiv \varphi$$

2. uzávěr formule $cl(\varphi) = \{ \underset{\substack{\parallel \\ \neg\neg p}}{p}, \neg p, (p \vee \neg p), \cancel{\neg(p \vee \neg p)}, \varphi, \neg\varphi \}$

nesplnitelné formule lze odstranit.

3. Výběr konzistentních podmnožin:

– každá podmnožina bude mít buď p nebo $\neg p$:

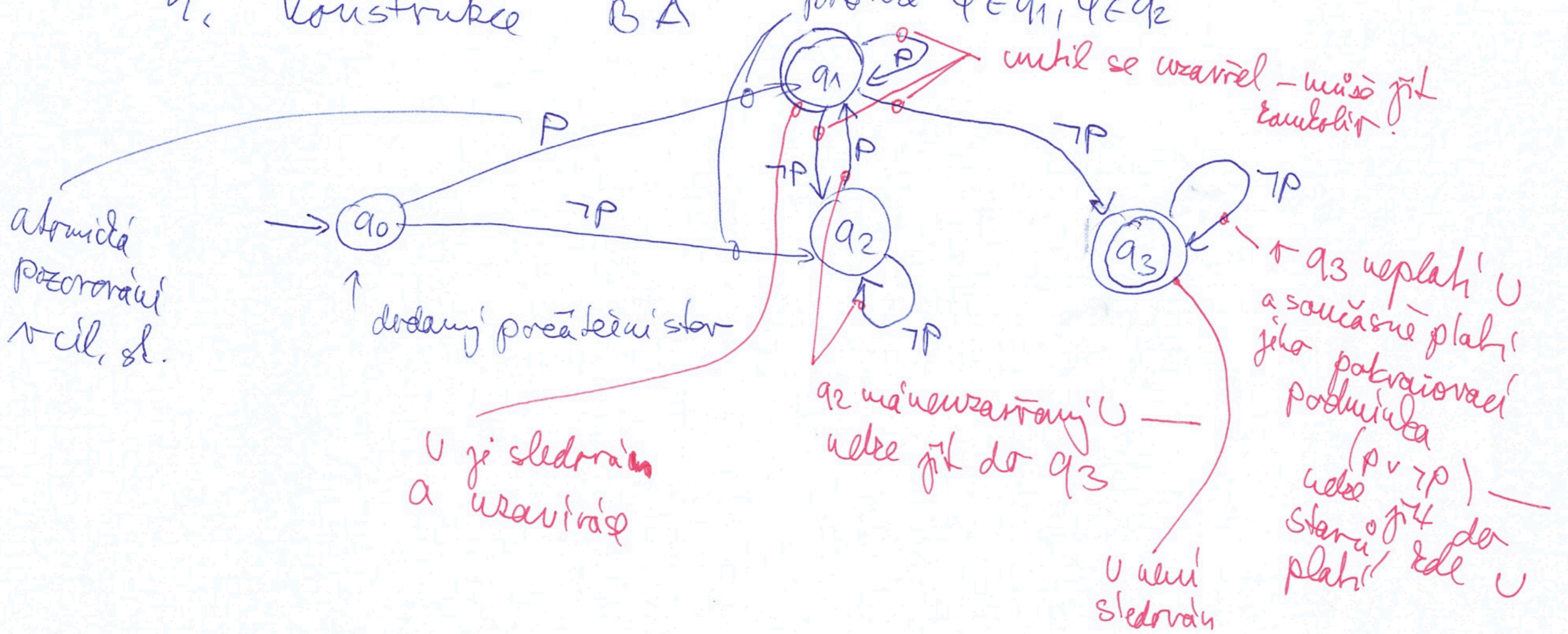
– $\underset{\substack{\parallel \\ q_1}}{\{} p, p \vee \neg p, \varphi \}$ – musí zde být, protože v množině je p ($\neg\varphi$ nelze – until se zde uzavírá)

↑ protože p je v dané množině

– $\underset{\substack{\parallel \\ q_2}}{\{} \neg p, p \vee \neg p, \varphi \}$

$\underset{\substack{\parallel \\ q_3}}{\{} \neg p, p \vee \neg p, \neg\varphi \}$ – obě možnosti jsou konzistentní!

1. konstrukce BA



4. Abstraktní interpretace: Nechť $\mathbb{F} = \{ f : \mathbb{A} \longrightarrow \mathbb{A} \}$ je množina všech parciálních f-cí nad množinou $\mathbb{A}$. Dále nechť $\mathbb{R} = 2^{\mathbb{A} \times \mathbb{A}}$ (mn-a všech binárních relací na $\mathbb{A}$). Vytvořte Galoisovo spojení, kosiční jehož jedné složkové komponenty bude mn-a $2^{\mathbb{F}}$ a druhým bude mn-a $\mathbb{R}$. Zdůvodněte, že se jedná opravdu o Galoisovo spojení. Zobrazení nesmí být konstantní.

– $(\underbrace{2^{\mathbb{F}}, \subseteq}) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} (\mathbb{R}, \underbrace{\subseteq})$

jeden prvek může vypadat např. takto:

$\{\{a \mapsto b, b \mapsto c, c \mapsto a\}, \{a \mapsto d, b \mapsto d, c \mapsto d\}, \dots\}$

jeden prvek

$\{(a,b), (a,d), (c,d), \dots\}$

$\alpha(F) = \bigcup_{f \in F} f$ pro $F \subseteq \mathbb{F}$

$\gamma(R) = \{f \in \mathbb{F} \mid f \subseteq R\}$ pro $R \in \mathbb{R}$

– Dokážeme, že se opravdu jedná o Galoisovo spojení!

– Ukážeme, že $\forall F \in 2^{\mathbb{F}} \; \forall R \subseteq \mathbb{R} \, . \; \alpha(F) \subseteq R \Leftrightarrow F \subseteq \gamma(R)$

– „$\Rightarrow$"

– Předp., že $\alpha(F) \subseteq R$.

– Potřebujeme ukázat, že pak také $F \subseteq \gamma(R)$.

– Ukážeme, že pro $\forall f \in F : f \in \gamma(R)$.

– Protože $f \in F$ a $\alpha(F) = \bigcup_{g \in F} g$, platí $f \subseteq \alpha(F)$

- z předp., že $\alpha(F) \subseteq R$, pak máme $f \subseteq R$
- $\mathcal{Y}(R)$ obsahují všechny f-ce g takové, že $g \subseteq R$.
- Tedy $f \in \mathcal{Y}(R)$.

- „$\Leftarrow$" Analogie (doma). $\square$

5. Sestavte v monotónním rámci analýzu toku dat, která pro každý řádek spolehlivě určí, které proměnné musí mít na daném řádku hodnotu 0. Uvažujeme pouze celočíselné proměnné z množiny V a příkazy:
    - read(&v), $v \in V$
    - $v_1 := v_2$, $v_1, v_2 \in V$
    - $v_1 := c$, $v_1 \in V$, $c \in \mathbb{Z}$
    - $v_1 := v_2 + v_3$
      $v_1 := v_2 * v_3$, $v_1, v_2, v_3 \in V$
    - $[v == 0]$ / $[v \neq 0]$ pro $v \in V$

Lze vést analýzu po jednotlivých příkazech s takové závislými gen/kill množinami. Na počátku jsou všechny proměnné vynulované!

- nosič $(\mathcal{L}, \sqsubseteq) = (2^V, \subseteq)$
- maximum $\top = V$
  minimum $\bot = \emptyset$
  průsek $\sqcap = \cap$
- směr analýzy: po směru
- extrémní tažová hodnota $In_{start} = V$
- tokové f-ce pro jednotlivé příkazy s:
  $f_s(X) = Gen_s(X) \cup (X \setminus Kills(X))$
- předávací f-ce z výstupu příkazu s na vstup dalšího s':
  $In_{s'} = \bigcap_{s \in pre(s')} Out_s$
  (platí i pro počáteční uzel, protože neutrální prvek vůči $\cap$ a $\subseteq$ je V.)
- Množiny $Gen_s(X)$ a $Kills(X)$:

	$Kills(X)$	$Gen_s(X)$
read(v)	$\{v\}$	$\emptyset$
$v_1 := v_2$	$\{v_1\}$	$(v_2 \in X)\ ?\ \{v_1\} : \emptyset$
$v_1 := c$	$\{v_1\}$	$(c = 0)\ ?\ \{v_1\} : \emptyset$

$v_1 := v_2 + v_3$	$\{v_1\}$	$(v_2 \in X \wedge v_3 \in X)\ ?\ \{v_1\} : \emptyset$
$v_1 := v_2 * v_3$	$\{v_1\}$	$(v_2 \in X \vee v_3 \in X)\ ?\ \{v_1\} : \emptyset$
$[v == 0]$	$\emptyset$	$\{v\}$
$[v \neq 0]$	$\{v\}$	$\emptyset$