

IV113 Validace a verifikace

Převod LTL formule na Büchi automat

doc. RNDr. Jiří Barnat, Ph.D.

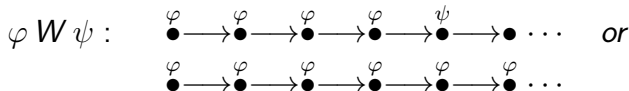
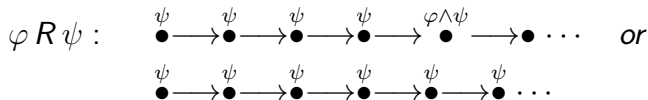
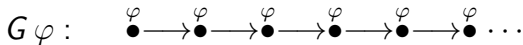
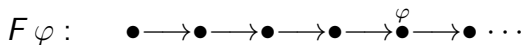
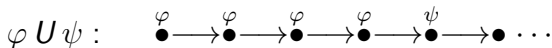
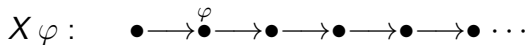
Problém

- Kripkeho struktura M
- LTL formule φ
- $M \models \varphi$?

Řešení pomocí Büchi automatů

- A_{sys} – automat akceptující běhy modelu
- $A_{\neg\varphi}$ – automat akceptující běhy porušující vlastnost φ
- $L(A_{sys}) \cap L(A_{\neg\varphi}) = L(A_{sys} \times A_{\neg\varphi})$
- $L(A_{sys} \times A_{\neg\varphi}) \neq \emptyset \iff$ model má běh porušující φ
- $M \models \varphi \iff A_{sys} \times A_{\neg\varphi}$ nemá akceptující cyklus

Grafické znázornění sémantiky temporálních operátorů



Nechť $\Sigma = \{a, b, c\}$, najděte Büchi automat, který akceptuje ω -regulární jazyk definovaný následující LTL formulí.

- a) $a U b$
- b) $a U (X b)$
- c) $\neg(a U (X b))$
- d) $a U (b U c)$
- e) $\neg(a U (b U c))$

Algoritmus převodu
LTL formule na Büchi automat

Vstup: Množina atomických propozic AP , LTL formule φ .
Výstup: Büchi automat A takový, že $L(A) = L_\varphi$.

Postup:

- Formule φ se převede do normální formy.
- Vypočítá se přechodový graf budoucího automatu.
- Graf se doplní na zobecněný Büchi automat.
- Zobecněný BA se převede na standardní Büchi automat.

Řekneme, že LTL formule je v **normální formě**, pokud neobsahuje operátory F a G , a všechny operátory unární negace jsou aplikovány na podformule tvořené pouze atomickou propozicí.

Syntax

$$\varphi ::= p \mid \neg p \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid X\varphi \mid \varphi U \psi \mid \varphi R \psi$$

Pravidla pro převod do normální formy

$$\neg(\varphi \vee \psi) \equiv (\neg\varphi) \wedge (\neg\psi)$$

$$\neg(\varphi \wedge \psi) \equiv (\neg\varphi) \vee (\neg\psi)$$

$$\neg X\varphi \equiv X(\neg\varphi)$$

$$\neg(\varphi U \psi) \equiv (\neg\varphi R \neg\psi)$$

$$\neg(\varphi R \psi) \equiv (\neg\varphi U \neg\psi)$$

Nechť $AP = \{a, b\}$. Převedte následující LTL formule do normální formy

a) $G(F(a))$

b) $F(G(a))$

c) $\neg(G(F(a)))$

d) $G(a \implies F(b))$

e) $\neg(a U (G b))$

Büchi automaty

- $A = (\Sigma, S, s, \delta, F)$
- $F \subseteq S$ je množina koncových stavů.
- Běh ρ je akceptující, pokud $\text{inf}(\rho) \cap F \neq \emptyset$.

Zobecněné Büchi automaty

- $A = (\Sigma, S, s, \delta, \mathcal{F})$
- $\mathcal{F} \subseteq 2^S$ je systém množin koncových stavů.
- Běh ρ je akceptující, pokud $\forall F_i \in \mathcal{F}$ platí $\text{inf}(\rho) \cap F_i \neq \emptyset$.

Nechť $\Sigma = \{a, b\}$ a $L = \{w \in \Sigma^\omega \mid \text{inf}(w) = \{a, b\}\}$.

Najděte zobecněný BA \mathcal{A} takový, že $L(\mathcal{A}) = L$.

Tvrzení

- Ke každému zobecněnému Büchi automatu A existuje (normální) Büchi automat B takový, že $L(A) = L(B)$.

Konstrukce

- Nechť $A = (\Sigma, S, s, \delta, \{F_1, \dots, F_n\})$.
- $B = (\Sigma, S \times \{0, \dots, n\}, (s, 0), \delta', S \times \{n\})$, kde
- $(q, y) \in \delta'((p, x), a)$ pokud $q \in \delta(p, a)$ a pro x a y platí
 - jestliže $q \in F_i$ a $x = i - 1$, tak $y = i$
 - jestliže $x = n$, tak $y = 0$
 - jinak $x = y$.

Příklad ZBA \rightarrow BA

ZBA: $\mathcal{F} = \{F_1 = \{p\}, F_2 = \{q\}\}$

	a	b
$\rightarrow p$	p	q
q	p	q

BA: $F = \{(p, 2), (q, 2)\}$

	a	b
$\rightarrow (p,0)$	(p,1)	(q,0)
(q,0)	(p,1)	(q,0)
(p,1)	(p,1)	(q,2)
(q,1)	(p,1)	(q,2)
$\leftarrow (p,2)$	(p,0)	(q,0)
$\leftarrow (q,2)$	(p,0)	(q,0)

Výpočet přechodového grafu

Pozorování

- Přechod v Büchi automatu stráží jeden stav běhu.
- Pro definici přechodu, je třeba vědět, co platí v aktuálním stavu, a co má platit v následujícím stavu běhu.

Rozbalené definice modálních operátorů

$$\begin{aligned} X a &\equiv tt \quad \wedge \quad X(a) \\ a U b &\equiv a \quad \wedge \quad X(a U b) \\ &\quad \vee \quad b \quad \wedge \quad X(tt) \\ a R b &\equiv b \quad \wedge \quad X(a R b) \\ &\quad \vee \quad a \wedge b \quad \wedge \quad X(tt) \end{aligned}$$

Pozorování

- Přechod v Büchi automatu stráží jeden stav běhu.
- Pro definici přechodu, je třeba vědět, co platí v aktuálním stavu, a co má platit v následujícím stavu běhu.

Rozbalené definice modálních operátorů

	Now	Next
$X a$	\equiv	a
$a U b$	$\equiv a$ $\vee b$	$a U b$
$a R b$	$\equiv b$ $\vee a \wedge b$	$a R b$

Uzel je uspořádaná pětice

- **Id** – Číslo
 - Unikátní označení uzlu.
- **Incomming** – Množina označení uzlů.
 - Množina přímých předchůdců vrcholu ve výsledném grafu.
 - Kóduje hrany grafu.
- **Now** – Množina LTL formulí.
 - Seznam podformulí, které platí v daném uzlu.
- **New** – Množina LTL formulí.
 - Množina ještě nezpracovaných formulí, které musí být splněny v tomto uzlu.
- **Next** – Množina LTL formulí.
 - Seznam formulí, které musí být splněny v následujícím uzlu.

Vytvoření grafu (výpočet množiny uzlů)

```
proc create_graph( $\varphi$ )  
  N = (new_ID(), {init},  $\emptyset$ , { $\varphi$ },  $\emptyset$ )  
  return expand(N,  $\emptyset$ )  
end
```

Pomocné funkce

- $expand(n, Nodes)$
 - Funkce volaná pro uzel n a dosud známe uzly $Nodes$.
 - Vrací množinu uzlů (po zpracování uzlu n).
- $new_ID()$
 - Každé volání této funkce vrátí dosud nevrácené číslo.
- $Neg(-)$
 - $Neg(A) = \neg A$ pro všechny $A \in AP$.
 - $Neg(True) = False$
 - $Neg(False) = True$
 - $\neg\neg A = A$

Graf LTL formule – funkce expand

```
proc expand(q, Nodes)
  if New(q) ==  $\emptyset$ 
    then if ( $\exists r \in \text{Nodes}$  takový, že  $\text{Now}(r) == \text{Now}(q) \wedge \text{Next}(r) == \text{Next}(q)$ )
      then  $\text{Incomming}(r) = \text{Incomming}(r) \cup \text{Incomming}(q)$ 
        return Nodes
      else  $N = (\text{new\_ID}(), \{ID(q)\}, \emptyset, \text{Next}(q), \emptyset)$ 
        return  $\text{expand}(N, \text{Nodes} \cup \{q\})$           /* q je nový uzel */
    fi
  else let  $\eta \in \text{New}(q)$ 
     $\text{New}(q) = \text{New}(q) \setminus \{\eta\}$ 
    if  $\eta \in \text{Now}(q)$                                 /*  $\eta$  již byla zpracována */
      then return  $\text{expand}(q, \text{Nodes})$ 
    fi
    switch ( $\eta$ )  /* pokračuj podle typu nejvnějšního operátoru  $\eta$  */
      ...
    end
  fi
end
```

Graf LTL formule – funkce expand (switch)

```
switch ( $\eta$ )          /* pokračuj podle typu nejnějnějšího operátoru  $\eta$  */  
  
  case ( $\eta \in (AP \cup \text{Neg}(AP) \cup \{True, False\})$ )  
    if ( $\eta == False \vee \text{Neg}(\eta) \in \text{Now}(q)$ )  
      then return Nodes  
      else  $N = (\text{new\_ID}(), \text{Incomming}(q), \text{Now}(q) \cup \{\eta\}, \text{New}(q), \text{Next}(q))$   
          return expand(N, Nodes)  
    fi  
  end  
  
  case ( $\eta \equiv \varphi U \psi$ )  
     $N1 = (\text{new\_ID}(), \text{Incomming}(q),$   
           $\text{Now}(q) \cup \{\eta\}, \text{New}(q) \cup \{\varphi\}, \text{Next}(q) \cup \{\varphi U \psi\})$   
     $N2 = (\text{new\_ID}(), \text{Incomming}(q),$   
           $\text{Now}(q) \cup \{\eta\}, \text{New}(q) \cup \{\psi\}, \text{Next}(q))$   
    return expand(N2, expand(N1, Nodes))  
  end  
  
  ...
```

Graf LTL formule – funkce expand (switch)

```
case ( $\eta \equiv \varphi R \psi$ )  
  N1 = (new_ID(), Incomming(q),  
        Now(q)  $\cup$  { $\eta$ }, New(q)  $\cup$  { $\varphi, \psi$ }, Next(q))  
  N2 = (new_ID(), Incomming(q),  
        Now(q)  $\cup$  { $\eta$ }, New(q)  $\cup$  { $\psi$ }, Next(q)  $\cup$  { $\varphi R \psi$ })  
  return expand(N2, expand(N1, Nodes))  
end
```

```
case ( $\eta \equiv \varphi \vee \psi$ )  
  N1 = (new_ID(), Incomming(q),  
        Now(q)  $\cup$  { $\eta$ }, New(q)  $\cup$  { $\varphi$ }, Next(q))  
  N2 = (new_ID(), Incomming(q),  
        Now(q)  $\cup$  { $\eta$ }, New(q)  $\cup$  { $\psi$ }, Next(q))  
  return expand(N2, expand(N1, Nodes))  
end
```

...

Graf LTL formule – funkce expand (switch)

```
case ( $\eta \equiv \varphi \wedge \psi$ )  
  N = (new_ID(), Incomming(q),  
      Now(q)  $\cup$  { $\eta$ }, New(q)  $\cup$  { $\varphi, \psi$ }, Next(q))  
  return expand(N, Nodes)  
end
```

```
case ( $\eta \equiv X \varphi$ )  
  N = (new_ID(), Incomming(q),  
      Now(q)  $\cup$  { $\eta$ }, New(q), Next(q)  $\cup$  { $\varphi$ })  
  return expand(N, Nodes)  
end
```

end

/* end of switch */

Příklad výpočet grafu pro formuli $X(a)$

Vypočtené uzly

Id:	2
Incomming:	init
Now:	$X(a)$
New:	\emptyset
Next:	a

Id:	4
Incomming:	2
Now:	a
New:	\emptyset
Next:	\emptyset

Id:	5
Incomming:	4,5
Now:	\emptyset
New:	\emptyset
Next:	\emptyset

Výpočet

Id	Incomming	Now	New	Next
1	init	\emptyset	$\{X(a)\}$	\emptyset
2	init	$\{X(a)\}$	\emptyset	$\{a\}$

Uzel 2 je nově vypočtený uzel.

3	2	\emptyset	$\{a\}$	\emptyset
4	2	$\{a\}$	\emptyset	\emptyset

Uzel 4 je nově vypočtený uzel.

5	4	\emptyset	\emptyset	\emptyset
---	---	-------------	-------------	-------------

Uzel 5 je nově vypočtený uzel.

6	5	\emptyset	\emptyset	\emptyset
---	---	-------------	-------------	-------------

Uzel 6 je shodný s uzlem 5.

$Incoming(5) = Incoming(5) \cup \{5\}$

Příklad $aU(bUc)$

01| in, \emptyset , $\{aU(bUc)\}$, \emptyset

02| in, $\{aU(bUc)\}$, $\{a\}$, $\{aU(bUc)\}$

04| in, $\{aU(bUc),a\}$, \emptyset , $\{aU(bUc)\}$

Uzel 04 je nově vypočtený uzel.

05| 04, \emptyset , $\{aU(bUc)\}$, \emptyset

06| 04, $\{aU(bUc)\}$, $\{a\}$, $\{aU(bUc)\}$

08| 04, $\{aU(bUc),a\}$, \emptyset , $\{aU(bUc)\}$

Uzlu 04 je přidán předchůdce 04.

07| 04, $\{aU(bUc)\}$, $\{bUc\}$, \emptyset

09| 04, $\{aU(bUc),bUc\}$, $\{b\}$, $\{bUc\}$

11| 04, $\{aU(bUc),bUc,b\}$, \emptyset , $\{bUc\}$

Uzel 11 je nově vypočtený uzel.

12| 11, \emptyset , $\{bUc\}$, \emptyset

13| 11, $\{bUc\}$, $\{b\}$, $\{bUc\}$

15| 11, $\{bUc,b\}$, \emptyset , $\{bUc\}$

Uzel 15 je nově vypočtený uzel.

16| 15, \emptyset , $\{bUc\}$, \emptyset

17| 15, $\{bUc\}$, $\{b\}$, $\{bUc\}$

19| 15, $\{bUc,b\}$, \emptyset , $\{bUc\}$

Uzlu 15 je přidán předchůdce 15.

03| in, $\{aU(bUc)\}$, $\{bUc\}$, \emptyset

07| 04, $\{aU(bUc)\}$, $\{bUc\}$, \emptyset

10| 04, $\{aU(bUc),bUc\}$, $\{c\}$, \emptyset

14| 11, $\{bUc\}$, $\{c\}$, \emptyset

18| 15, $\{bUc\}$, $\{c\}$, \emptyset

Příklad $a U (b U c)$ – pokračování

18 | 15, {bUc}, {c}, \emptyset

20 | 15, {bUc,c}, \emptyset , \emptyset

Uzel 20 je nově vypočtený uzel.

21 | 20, \emptyset , \emptyset , \emptyset

Uzel 21 je nově vypočtený uzel.

22 | 21, \emptyset , \emptyset , \emptyset

Uzlu 21 je přidán předchůdce 21.

14 | 11, {bUc}, {c}, \emptyset

23 | 11, {bUc,c}, \emptyset , \emptyset

Uzlu 20 je přidán předchůdce 11.

10 | 04, {aU(bUc),bUc}, {c}, \emptyset

24 | 04, {aU(bUc),bUc,c}, \emptyset , \emptyset

Uzel 24 je nově vypočtený uzel.

25 | 24, \emptyset , \emptyset , \emptyset

Uzlu 21 je přidán předchůdce 24.

Příklad $a U (b U c)$ – pokračování

03| in, $\{aU(bUc)\}$, $\{bUc\}$, \emptyset

26| in, $\{aU(bUc),bUc\}$, $\{b\}$, $\{bUc\}$

28| in, $\{aU(bUc),bUc,b\}$, \emptyset , $\{bUc\}$

Uzlu 11 je přidán předchůdce *in*.

27| in, $\{aU(bUc),bUc\}$, $\{c\}$, \emptyset

27| in, $\{aU(bUc),bUc\}$, $\{c\}$, \emptyset

29| in, $\{aU(bUc),bUc,c\}$, \emptyset , \emptyset

Uzlu 24 je přidán předchůdce *in*.

Předpoklady

- Dána množina AP.
- *Nodes* je množina vrcholů grafu LTL formule.

Zobecněný Büchi automat $A = (S, \Sigma, \delta, init, \mathcal{F})$

- $S = Nodes \cup \{init\}$
- $\Sigma = 2^{AP}$
- $r' \in \delta(r, \alpha)$ pokud
 - $r \in Incomming(r'), \alpha \in \Sigma$
 - α splňuje omezení dané množinou $((AP \cup \neg AP) \cap Now(r'))$
- $\mathcal{F} = \{F_1, \dots, F_n\}$
 - Pro každou podformuli ve tvaru $\varphi U \psi$ definujeme F_i .
 - $F_i = \{r \in Nodes \mid \psi \in Now(r) \vee \varphi U \psi \notin Now(r)\}$.

Přechodová funkce (stráže)

	a	b	c	tt
→ init	04	11	24	
04	04	11	24	
11		15	20	
15		15	20	
20				21
21				21
24				21

\mathcal{F} – akceptující množiny

- $F_{aU(bUc)} = \{11, 15, 20, 21, 24\}$
- $F_{bUc} = \{04, 20, 21, 24\}$