Verificação Formal de Software

Nelma Moreira

Departamento de Ciência de Computadores da FCUP

Verificação Formal de Software
Aula 13

Autómato Alternado de Büchi

Seja $A = (\Sigma, S, s^0, \delta, F)$ um autómato alternado em palavras infinitas, onde $s^0 \in S$, estado inicial ,$F \subseteq S$ estados finais, $\delta : S \times \Sigma \to \mathcal{B}^+(S)$.

Computação

Uma computação de A numa palavra infinita $w = a_0 a_1 \ldots$ é uma árvore S-etiquetada r tal que $r(\epsilon) = s^0$ e:
se $|x| = i$, $r(x) = s$ e $\delta(s, a_i) = \theta$, então x tem k filhos $x_1, \ldots, x_k$ com $k \leq |S|$ e $\{r(x_1), \ldots r(x_k)\}$ satisfaz θ.
Uma computação é de aceitação se todo o ramo infinito de r inclui um número infinito de etiquetas de F. Se $\delta(s, a_i) = \top$, x não tem filhos . Uma computação finita é de aceitação.

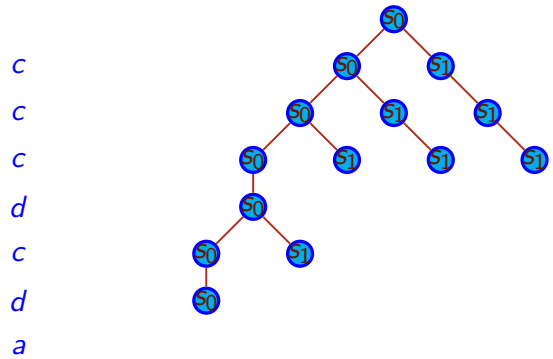
$$L_\omega(A) = \{w \in \Sigma^\omega \mid \text{existe uma computação de aceitação } r \text{ para } w\}$$

Exemplo

$\mathcal{A} = (\{a, b, c, d\}, \{s_0, s_1, s_2\}, \delta, s_0, \{s_1\})$

δ	a	b	c	d
s_0	$\top$	$\top$	$s_0 \wedge (s_1 \vee s_2)$	s_0
s_1	$\bot$	$\top$	s_1	$\top$
s_2	$\top$	s_1	$\top$	s_2

- Determina uma computação sobre a palavra *cccdcdad* . . .
- Determina uma computação de aceitação e outra de não aceitação para a palavra *caccccc* . . .


c
c
c
d
c
d
a
S0
S0
S1
S0
S1
S1
S0
S1
S1
S1
S0
S0
S1
S0

Autómato Alternado de Büchi

Teorema

Uma linguagem é aceite por um autómato alternado de Büchi se e só se for aceite por um autómato não determinístico de Büchi.

Complementar

Complementação de autómatos alternados de Büchi não é fácil: não passar um número infinito de vezes por estados finais, não é o mesmo que passar um número infinito de vezes por estados não finais.

Teorema

O problema de determinar se a linguagem aceite por um autómatos alternado de Büchi é vazia é decidível em tempo exponencial.

Satisfazibilidade do LTL

Satisfazibilidade

Dado um modelo $\mathcal{M} = (S, \rightarrow, L)$ e um caminho $\pi = s_1 \rightarrow \ldots$, define-se a relação de satisfazibilidade $\models$ indutivamente por:

1. $\pi \models p$ sse $p \in L(s_1)$
2. $\pi \models \neg\phi$ sse $\pi \not\models \phi$
3. $\pi \models \phi \wedge \psi$ sse $\pi \models \phi$ e $\pi \models \psi$
4. $\pi \models \mathsf{X}\phi$ sse $\pi^2 \models \phi$
5. $\pi \models \phi \mathsf{U} \psi$ sse $\exists i \geq 1, \pi^i \models \psi$ e $\forall\, 1 \leq j < i,\ \pi^j \models \phi$

Dado L, um caminho pode ser visto como uma palavra infinita sobre $\Sigma = 2^{\mathtt{Atoms}}$. Se $a \in \Sigma$, a é uma valorização que torna verdadeiras as variáveis proposicionais em a.

Vamos ver que o conjunto de caminhos que satisfazem uma fórmula são os aceites por um autómato sobre palavras infinitas.

LTL e Autómatos de Büchi

Considerando o alfabeto $\Sigma = 2^{\mathtt{Atoms}}$ não é difícil associar a fórmulas do LTL autómatos de Büchi... mas é muito pouco eficiente computacionalmente.

Para ϕ fórmula proposicional seja

$$\Sigma_\phi = \{a \in \Sigma \mid a \models \phi\}$$

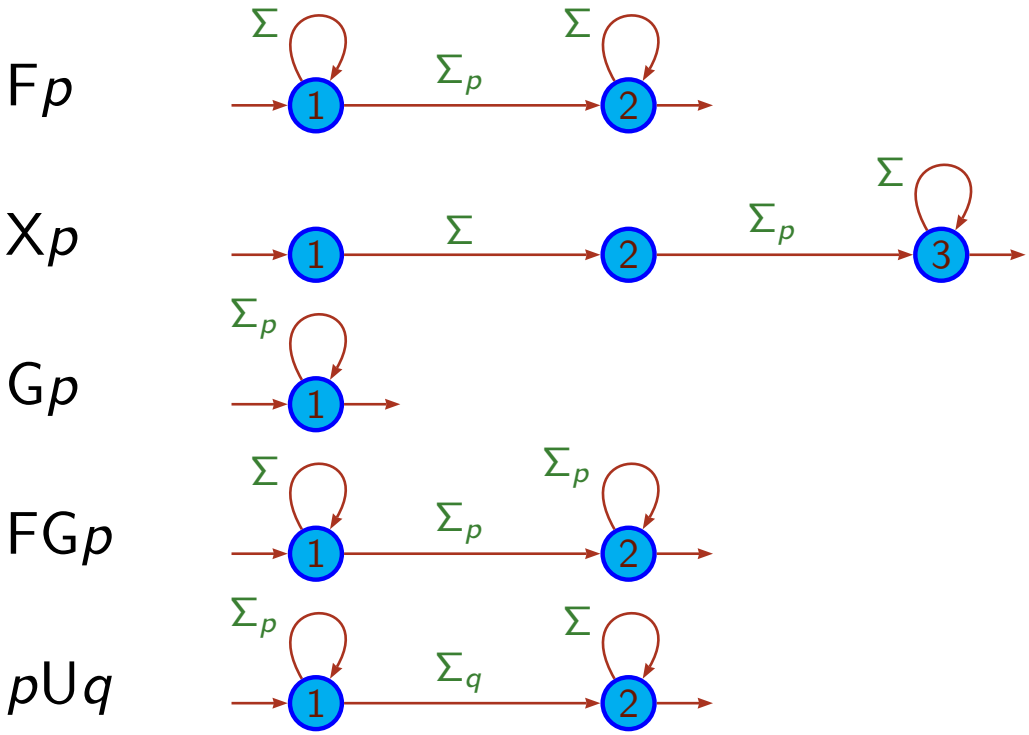
.

Por exemplo, se $p \in \mathit{Atoms}$, $\Sigma_p = \{a \in \Sigma \mid p \in a\}$, $\Sigma_{\neg p} = \Sigma \setminus \Sigma_p$, $\Sigma_{p \wedge q} = \Sigma_p \cap \Sigma_q$, $\Sigma_{p \vee q} = \Sigma_p \cup \Sigma_q$.

Temos que dados estados s, s'

$$s \xrightarrow{\Sigma_\phi} s' = \{s \xrightarrow{a} s' \mid a \in \Sigma_\phi\}$$

LTL e Autómatos de Büchi



Como estes autómatos são fechados para a reunião, interseção e complementar, podem ser construídos para qualquer fórmula. Mas os autómatos para a negação levam a uma explosão combinatória...

LTL e Autómatos Alternados de Büchi

Teorema

Dada uma fórmula ϕ do LTL podemos construir um autómato alternado de Büchi $A_\phi = (2^{\mathtt{Atoms}}, S, \phi, \delta, F)$ tal que $|S| = \mathcal{O}(|\phi|)$ e tal que $L_\omega(A_\phi)$ é exactamente o conjunto de caminhos que satisfazem a fórmula ϕ.

Demonstração.

Seja S o conjunto de todas as subfórmulas de ϕ e das suas negações . Seja F o conjunto de todas as fórmulas de S da forma $\neg(\psi \mathsf{U} \phi)$. Considere-se o dual duma fórmula, estendido a negações $\overline{\neg\phi} = \phi$ e $\overline{\phi} = \neg\phi$. Então δ:

- $\delta(p, a) = \top$ se $p \in a$
- $\delta(p, a) = \bot$ se $p \notin a$
- $\delta(\phi \wedge \psi, a) = \delta(\phi, a) \wedge \delta(\psi, a)$
- $\delta(\neg\phi, a) = \overline{\delta(\phi, a)}$
- $\delta(\mathsf{X}\phi, a) = \phi$
- $\delta(\phi \mathsf{U} \psi, a) = \delta(\psi, a) \vee (\delta(\phi, a) \wedge \phi \mathsf{U} \psi)$

LTL e Autómatos Alternados de Büchi

Seja r uma computação de A_ϕ. Os ramos infinitos a partir de certa altura ou são etiquetados por $\phi \mathsf{U} \psi$ ou por $\neg(\phi \mathsf{U} \psi)$. Temos que

$$\delta(\neg(\phi \mathsf{U} \psi), a) = \overline{\delta(\psi, a)} \ \wedge \ (\overline{\delta(\phi, a)} \ \vee \ \neg(\phi \mathsf{U} \psi))$$

então se um ramo for etiquetado por $\neg(\phi \mathsf{U} \psi)$ isso garante que $\phi \mathsf{U} \psi$ não se verifica a partir daí (porque ψ não é satisfeito) . Por isso esses são estados finais.

Por outro lado para os ramos etiquetados por $\phi \mathsf{U} \psi$ não é garantido que a $\phi \mathsf{U} \psi$ se verifique pois não há garantia que ψ vá ser satisfeito.

A demonstração do teorema segue por indução na estrutura da fórmula ϕ.

Corolário

Dada uma fórmula ϕ do LTL pode-se construir um autómato de Büchi A_ϕ tal que $L(A_\phi)$ é exactamente o conjunto de caminhos que satisfazem a fórmula ϕ.

Exercício

Para $\phi = Xp$, determina o autómato alternado de Büchi A_ϕ. Determina a conjunto de caminhos aceites por A_ϕ

$\diamond$

Exemplo

Seja $\phi = (\mathsf{X}\neg p)\mathsf{U}q$.

$$A_\phi = (2^{\{p.q\}}, \{\phi, \neg\phi, \mathsf{X}\neg p, \neg\mathsf{X}\neg p, \neg p, p, \neg q, q\}, \phi, \delta, \{\neg\phi\})$$

s	$\{p,q\}$	$\{p\}$	$\{q\}$	$\emptyset$
ϕ	$\top$	$\neg p \wedge \phi$	$\top$	$\neg p \wedge \phi$
$\neg\phi$	$\bot$	$p \vee \neg\phi$	$\bot$	$p \vee \neg\phi$
$\mathsf{X}\neg p$	$\neg p$	$\neg p$	$\neg p$	$\neg p$
$\neg\mathsf{X}\neg p$	p	p	p	p
$\neg p$	$\bot$	$\bot$	$\top$	$\top$
p	$\top$	$\top$	$\bot$	$\bot$
q	$\top$	$\bot$	$\top$	$\bot$
$\neg q$	$\bot$	$\top$	$\bot$	$\top$

No estado ϕ se q não se verifica então $\neg p$ e ϕ tem de se verificar no estado seguinte.Como $\phi \notin F$, A_ϕ terá de chegar a um estado tal que q se verifica.

Modelos e Autómatos de Büchi

Seja um sistema de transições (modelo) $M = (S, \rightarrow, L)$, para um conjunto de variáveis proposicionais $\mathtt{Atoms}$. Um caminho $\pi = s_0 s_1 \ldots$ pode ser visto como uma sequência de subconjuntos de $\mathtt{Atoms}$.

Dado um modelo M e $s_0 \in S$, associamos um autómato de Büchi

$$A_M = (2^{\mathtt{Atoms}}, S, \{s_0\}, \delta, S)$$

, tal que $s' \in \delta(s, a)$ sse $s \rightarrow s'$ e $a = L(s)$.

Como o conjunto de estados finais coincide com S, qualquer caminho π é uma computação de aceitação e então $L_\omega(A_M)$ coincide com o conjunto de caminhos de M.

Exercícios

Exercício

Considera o modelo $\mathcal{M} = (S, \rightarrow, L)$ *com*

- $S = \{q_1, q_2, q_3, q_4\}$,
- $\rightarrow = \{q_1 \rightarrow q_2, q_1 \rightarrow q_4, q_2 \rightarrow q_4, q_3 \rightarrow q_2, q_3 \rightarrow q_3, q_4 \rightarrow q_4\}$,
- *e* $L(q_1) = \{\}, L(q_2) = \{b\}, L(q_3) = \{a\}, L(q_4) = \{a, b\}$).

a) *Representa* $(\mathcal{M}, q_3)$ *por um autómato não determinístico de Büchi* $\mathcal{A}_{(\mathcal{M},q_3)}$.

b) *Mostra que existe uma palavra* $\omega \in L(\mathcal{A}_{(\mathcal{M},q_3)}) \cap L(A_{\neg(aUb)})$ *(ver alínea a) do exercício anterior).*

c) *O que podes concluir acerca de* $\mathcal{M}, q_3 \models aUb$*?*

$\diamond$

Algoritmo de *Model Checking* para o LTL

O problema de verificação de que um modelo satisfaz uma fórmula ϕ, $M, s_o \models \phi$ reduz-se a saber se

$$L_\omega(A_M) \subseteq L_\omega(A_\phi)$$

Ou equivalentemente,

$$L_\omega(A_M) \cap L_\omega(\overline{A_\phi}) = \emptyset$$

Notar que $L_\omega(\overline{A_\phi}) = L_\omega(A_{\neg\phi})$.

O autómato para a intersecção tem $|S|.2^{\mathcal{O}(|\phi|)}$ estados.

Logo o model checking pode ser feito em tempo $O(|S|.2^{\mathcal{O}(|\phi|)})$. Como a especifcação é em geral pequena, este algoritmo é razoavelmente eficiente...

Exercícios

Exercício

Considera o autómato alternado de Büchi
$\mathcal{A} = (\{a, b, c, d\}, \{s_0, s_1, s_2, s_3, s_4\}, \delta, s_0, \{s_4\})$, *onde*

δ	a	b	c	d
s_0	$s_2 \wedge s_3$	$s_1 \wedge s_3$	$s_1 \vee s_2$	s_0
s_1	s_4	s_1	s_1	s_0
s_2	s_2	s_4	s_2	s_0
s_3	s_3	s_3	s_3	s_0
s_4	s_4	s_4	s_4	s_0

Indica palavras ω_1 *e* ω_2 *tal que* $\omega_1 \in L(\mathcal{A})$ *e* $\omega_2 \notin L(\mathcal{A})$ *juntamente com uma computação de aceitação para* ω_1 *e uma de não aceitação para* ω_2. *Descreve informalmente* $L(\mathcal{A})$.

a) *Para* $\phi = a\ U\ b$ *determina o autómato alternado de Büchi* $A_{\neg\phi}$. *Determina o conjunto de caminhos aceites por* $A_{\neg\phi}$.

b) *Repite a alínea anterior para a fórmula* $a \wedge Fb$ *(lembra-te de que* $F\phi \equiv \top U\phi$*).*

◇

Bibliografia


M. Vardi.
An automata-theoretic approach to linear temporal logic.
In *Banff'94*, 1994.


Moshe Vardi.
Automata-theoretic techniques for temporal reasoning.
In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*. Elsevier, 2006.


M. Vardi and T. Wilke.
Automata: From logics to algorithms.
In *WAL 07*, pages 645–753, 2007.