Automates d'arbres

http://list.mpri.master.univ-paris7.fr/wws/info/cours-1-18

cours-1-18@mpri.master.univ-paris7.fr

Cours: Florent Jacquemard
TD: Ştefan Ciobâcă

INRIA Saclay & LSV (CNRS-ENS Cachan)

florent.jacquemard@inria.fr
ciobaca@lsv.ens-cachan.fr

2 novembre 2009

Logique monadique faible du second ordre à deux successeurs

Weak Second Order Monadic Logic with two successors

(WS2S)

Bibliographie

- **TATA book** (Tree Automata Theory and Application)
  Hubert Comon, Max Dauchet, Remi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, Marc Tommasi
  `http://tata.gforge.inria.fr`

  chapter 3 :
  - automata for tuples of trees,
  - logic WSkS,
  - applications.

Logic and Automata

- logic express properties of labeled binary trees
  = language specification,
- compilation of formulae into automata
  = decision algorithms.
- equivalence between both formalisms
  [Thatcher & Wright's theorem].

Interpretation Structures

$\mathcal{L}$:= set of predicate symbols $P_1, \ldots P_n$ with arity.

A structure $\mathcal{M}$ over $\mathcal{L}$ is a tuple

$$\mathcal{M} := \left\langle \mathcal{D}, P_1^{\mathcal{M}}, \ldots, P_n^{\mathcal{M}} \right\rangle$$

where

- $\mathcal{D}$ is the domain of $\mathcal{M}$,
- every $P_i^{\mathcal{M}}$ (interpretation of P_i) is a subset of $\mathcal{D}^{arity(P_i)}$.

Term as structure

Σ signature, k = maximal arity.

$$\mathcal{L}_\Sigma := \{=, <, S_1, \ldots, S_k, L_a \mid a \in \Sigma\}.$$

to $t \in \mathcal{T}(\Sigma)$, we associate a structure $\underline{t}$ over $\mathcal{L}_\Sigma$

$$\underline{t} := \left\langle \mathcal{P}os(t), =, <, S_1, \ldots, S_k, L_a^{\underline{t}}, L_b^{\underline{t}}, \cdots \right\rangle$$

where

- domain = positions of t $\quad (\mathcal{P}os(t) \subset \{1, \ldots, k\}^*)$
- $=$: equality over $\mathcal{P}os(t)$,
- $<$: prefix ordering over $\mathcal{P}os(t)$,
- $S_i = \{\langle p, p \cdot i\rangle \mid p, p \cdot i \in \mathcal{P}os(t)\}$ (i^{th} successor position),
- $L_a^{\underline{t}} = \{p \in \mathcal{P}os(t) \mid t(p) = a\}$.

FOL with k successors

- first order variables $x, y \ldots$
- form $::=$ $x = y \mid x < y$
  $\mid S_1(x, y) \mid \ldots \mid S_k(x, y) \mid L_a(x) \quad a \in \Sigma$
  $\mid$ form $\wedge$ form $\mid$ form $\vee$ form $\mid$ $\neg$form
  $\mid \exists x$ form $\mid \forall x$ form

Notation : $\phi(x_1, \ldots, x_m)$,
where $x_1, \ldots, x_m$ are the free variables of ϕ.

WSkS : syntax

- first order variables $x, y \dots$
- second order variables $X, Y \dots$
- $$\begin{array}{rcl} \text{form} & ::= & x = y \mid x < y \mid x \in X \\ & & \mid S_1(x,y) \mid \dots \mid S_k(x,y) \mid L_a(x) \quad a \in \Sigma \\ & & \mid \text{form} \wedge \text{form} \mid \text{form} \vee \text{form} \mid \neg\text{form} \\ & & \mid \exists x\, \text{form} \mid \exists X\, \text{form} \mid \forall x\, \text{form} \mid \forall X\, \text{form} \end{array}$$

Notation : $\phi(x_1, \dots, x_m, X_1, \dots, X_n)$,
where $x_1, \dots, x_m$, $X_1, \dots, X_n$ are the free variables of ϕ.

WSkS : semantics

- $t \in \mathcal{T}(\Sigma)$,
- valuation σ of first order variables into $\mathcal{P}os(t)$,
- valuation δ of second order variables into subsets of $\mathcal{P}os(t)$,
- $\underline{t}, \sigma, \delta \models x = y$ iff $\sigma(x) = \sigma(y)$,
- $\underline{t}, \sigma, \delta \models x < y$ iff $\sigma(x) <_{\mathit{prefix}} \sigma(y)$,
- $\underline{t}, \sigma, \delta \models x \in X$ iff $\sigma(x) \in \delta(X)$,
- $\underline{t}, \sigma, \delta \models S_i(x, y)$ iff $\sigma(y) = \sigma(x) \cdot i$,
- $\underline{t}, \sigma, \delta \models L_a(x)$ iff $t(\sigma(x)) = a$ i.e. $\sigma(x) \in L_a^t$,
- $\underline{t}, \sigma, \delta \models \phi_1 \wedge \phi_2$ iff $\underline{t}, \sigma, \delta \models \phi_1$ and $\underline{t}, \sigma, \delta \models \phi_2$,
- $\underline{t}, \sigma, \delta \models \phi_1 \vee \phi_2$ iff $\underline{t}, \sigma, \delta \models \phi_1$ or $\underline{t}, \sigma, \delta \models \phi_2$,
- $\underline{t}, \sigma, \delta \models \neg\phi$ iff $\underline{t}, \sigma, \delta \not\models \phi$,

WSkS : semantics (quantifiers)

- $\underline{t}, \sigma, \delta \models \exists x\ \phi$ iff $x \notin dom(\sigma)$, x free in ϕ and exists $p \in \mathcal{P}os(t)$ s.t. $\underline{t}, \sigma \cup \{x \mapsto p\}, \delta \models \phi$,
- $\underline{t}, \sigma, \delta \models \forall x\ \phi$ iff $x \notin dom(\sigma)$, x free in ϕ and for all $p \in \mathcal{P}os(t)$, $\underline{t}, \sigma \cup \{x \mapsto p\}, \delta \models \phi$,
- $\underline{t}, \sigma, \delta \models \exists X\ \phi$ iff $X \notin dom(\delta)$, X free in ϕ and exists $P \subseteq \mathcal{P}os(t)$ s.t. $\underline{t}, \sigma, \delta \cup \{X \mapsto P\} \models \phi$,
- $\underline{t}, \sigma, \delta \models \forall X\ \phi$ iff $X \notin dom(\delta)$, X free in ϕ and for all $P \subseteq \mathcal{P}os(t)$, $\underline{t}, \sigma, \delta \cup \{X \mapsto P\} \models \phi$.

WSkS : languages

Definition : WSkS-definability

For $\phi \in$ WSkS without free variables over $\mathcal{L}_\Sigma$,
$L(\phi) := \{t \in \mathcal{T}(\Sigma) \mid \underline{t} \models \phi\}$.

Example :

$\Sigma = \{a : 2, b : 2, c : 0\}$. Formula for the language of terms in $\mathcal{T}(\Sigma)$ containing the pattern $a(b(x_1, x_2), x_3)$.

Example :

Formula for the language of terms in $\mathcal{T}(\Sigma)$ such that all a-labelled node has a b-labelled child.

Example :

Formula for the language of terms in $\mathcal{T}(\Sigma)$ such that all a-labelled node has a b-labelled descendant.

WSkS : examples

- root position : root$(x) \equiv \neg\exists y\ y < x$
- inclusion : $X \subseteq Y \equiv \forall x (x \in X \Rightarrow x \in Y)$
- intersection : $Z = X \cap Y \equiv \forall x\ (x \in Z \Leftrightarrow (x \in X \wedge x \in Y))$
- emptiness : $X = \emptyset \equiv \forall x\ x \notin X$
- finite union :
$$X = \bigcup_{i=1}^{n} X_i \equiv \Big(\bigwedge_{i=1}^{n} X_i \subseteq X\Big) \wedge \forall x\ \Big(x \in X \Rightarrow \bigvee_{i=1}^{n} x \in X_i\Big)$$
- partition :

$$X_1, \ldots, X_n \text{ partition } X \equiv X = \bigcup_{i=1}^{n} X_i \wedge \bigwedge_{i=1}^{n-1} \bigwedge_{j=i+1}^{n} X_i \cap X_j = \emptyset$$

WS2S : examples (2)

- singleton :
  $\mathrm{sing}(X) \equiv X \neq \emptyset \wedge \forall Y \; \big(Y \subseteq X \Rightarrow (Y = X \vee Y = \emptyset)\big)$
- $\leq$ (without $<$)

$$x \leq y \quad \equiv \quad \forall X \left(\begin{array}{l} y \in X \\ \wedge\, \forall z \, \forall z' \, (z' \in X \wedge \bigvee_{i \leq k} S_i(z, z')) \Rightarrow z \in X \end{array} \right) \\ \Rightarrow x \in X$$

or

$$x \leq y \quad \equiv \quad \exists X \; \big(\forall z \; z \in X \Rightarrow (\exists z' \bigvee_{i \leq k} S_i(z', z) \wedge z' \in X) \vee z = x\big) \\ \wedge\, y \in X$$

Thatcher & Wright's Theorem

Theorem : Thatcher and Wright

Languages of WSkS formula = regular tree languages.

pr.: 2 directions (2 constructions) :

- TA $\rightarrow$ WSkS,
- WSkS $\rightarrow$ TA.

automates $\longrightarrow$ logique

regular languages $\rightarrow$ WSkS languages

Let $\Sigma = \{a_1, \ldots, a_n\}$.

Theorem :

For all tree automaton $\mathcal{A}$ over Σ, there exists $\phi_{\mathcal{A}} \in$ WSkS such that $L(\phi_A) = L(\mathcal{A})$.

$\mathcal{A} = (\Sigma, Q, Q^{\mathsf{f}}, \Delta)$ avec $Q = \{q_0, \ldots, q_m\}$.
$\phi_{\mathcal{A}}$: existence of an accepting run of $\mathcal{A}$ on $t \in \mathcal{T}(\Sigma)$.

$$\phi_{\mathcal{A}} := \exists Y_0 \ldots \exists Y_m \; \phi_1(\overline{Y}) \wedge \phi_2(\overline{Y}) \wedge \phi_3(\overline{Y}) \wedge \phi_4(\overline{Y})$$

regular languages $\to$ WSkS languages

$\phi_1(\overline{Y})$: every position is labeled with at most one state.

$$\phi_1(\overline{Y}) \equiv \bigwedge_{\substack{0 \leq i,j \leq m \\ i \neq j}} \forall x\ x \in Y_i \Rightarrow \neg x \in Y_j$$

$\phi_2(\overline{Y})$: the root is labeled with a final state

$$\phi_2(\overline{Y}) \equiv \forall x_0\ \mathsf{root}(x_0) \Rightarrow \bigvee_{q_i \in Q^{\mathsf{f}}} x_0 \in Y_i$$

regular languages $\rightarrow$ WSkS languages

$\phi_3(\overline{Y})$: transitions for constants symbols

$$\phi_3(\overline{Y}) \equiv \bigwedge_{a \in \Sigma_0} \Big(\forall x \; L_a(x) \Rightarrow \bigvee_{a \rightarrow q_i \in \Delta} x \in Y_i \Big)$$

$\phi_4(\overline{Y})$: transitions for non-constant symbols

$$\begin{array}{rcl} \phi_4(\overline{Y}) & \equiv & \displaystyle\bigwedge_{f \in \Sigma_j, 0 < j \leq k} \forall x \, \forall y_1 \ldots \forall y_j \\ & & \big(L_f(x) \wedge S_1(x, y_1) \wedge \ldots \wedge S_j(x, y_j) \big) \\ & & \Downarrow \\ & & \displaystyle\bigvee_{f(q_{i_1}, \ldots, q_{i_j}) \rightarrow q_i \in \Delta} x \in Y_i \wedge y_1 \in Y_{i_1} \wedge \ldots \wedge y_j \in Y_{i_j} \end{array}$$

FOL

Proposition :

The language L of terms with an even number of nodes labeled by a is regular (hence WSkS-definable) but not FOL-definable.

pr.: Ehrenfeucht-Fraïssé games.

logique $\longrightarrow$ automates

Theorem Thatcher & Wright

Theorem :

Every WSkS language is regular.

For all formula $\phi \in$ WSkS over Σ (without free variables) there exists a tree automaton $\mathcal{A}_\phi$ over Σ, such that $L(\mathcal{A}_\phi) = L(\phi)$.

Corollary :

WSkS is decidable.

pr.: reduction to emptiness decision for $\mathcal{A}_\phi$.

Theorem Thatcher & Wright

$\mathcal{A}_\phi$ is effectively constructed form ϕ, by induction.

- automata for atoms

  $\Rightarrow$ automata for formula with free variables.
- Boolean closures for Boolean connectors.
- $\exists$ quantifier : projection.

Theorem Thatcher & Wright

$\mathcal{A}_\phi$ is effectively constructed form ϕ, by induction.

✓ for free second order variables :

$$\frac{\begin{array}{c} t \in \mathcal{T}(\Sigma) \\ \delta : \{X_1, \ldots, X_n\} \to 2^{\mathcal{P}os(t)} \end{array} \quad \mapsto \quad t \times \delta \in \mathcal{T}(\Sigma \times \{0,1\}^n)}{}$$

arity of $\langle a, \overline{b} \rangle$ in $\Sigma \times \{0,1\}^n =$ arity of a in Σ.

for all $p \in \mathcal{P}os(t)$, $(t \times \delta)(p) = \langle t(p), b_1, \ldots, b_n \rangle$ where

for all $i \leq n$,

- $b_i = 1$ if $p \in \delta(X_i)$,
- $b_i = 0$ otherwise.

✓ free first order variables are interpreted as singletons.

$\mathsf{WS}k\mathsf{S}_0$

We consider a simplified language (wlog).

- no first order variables,
- only second order variables $X, Y \ldots$,
- $$\begin{array}{rcl} \text{form} & ::= & X \subseteq Y \mid Y = X \cdot 1 \mid \ldots \mid Y = X \cdot k \\ & \mid & X \subseteq L_a \quad a \in \Sigma \\ & \mid & \text{form} \wedge \text{form} \mid \text{form} \vee \text{form} \mid \neg\text{form} \\ & \mid & \exists X\, \text{form} \mid \forall X\, \text{form} \end{array}$$

interpretation $Y = X \cdot i$: $X = \{x\}$, $Y = \{y\}$ and $y = x \cdot i$.

ex : singleton

$$\text{singleton}(X) \equiv \exists Y \quad \big(Y \subseteq X \wedge Y \neq X \wedge \neg\exists Z\, (Z \subseteq X \wedge Z \neq X \wedge Z \neq Y)\big)$$

WSkS $\rightarrow$ WSkS$_0$

Lemma :

For all formula $\phi(x_1, \ldots, x_m, X_1, \ldots, X_n) \in$ WSkS,
there exists a formula $\phi'(X'_1, \ldots, X'_m, X_1, \ldots, X_n) \in$ WSkS$_0$
s.t. $\underline{t}, \sigma, \delta \models \phi(x_1, \ldots, x_m, X_1, \ldots, X_n)$
iff $\underline{t}, \sigma' \cup \delta \models \phi'(X'_1, \ldots, X'_m, X_1, \ldots, X_n)$, with $\sigma' : X'_i \mapsto \{\sigma(x_i)\}$.

pr.: several steps of formula rewriting :

1. elimination of $<$,
2. elimination of $S_1, \ldots, S_k$,
3. elimination of first order variables.

compilation of WSkS$_0$ into automata

notation : $\Sigma_{[m]} := \Sigma \times \{0,1\}^m$.

For all $\phi(X_1,\ldots,X_n) \in$ WSkS$_0$ and $m \geq n$,
we construct a tree automaton $[\![\phi]\!]_m$ over $\Sigma_{[m]}$ recognizing

$$\left\{t \times \delta \mid \delta : \{X_1,\ldots,X_m\} \to 2^{\mathcal{P}os(t)},\ \underline{t},\delta \models \phi(X_1,\ldots,X_n)\right\}$$

projection, cylindrification

projection

$proj_n : \quad \bigcup_{m \geq n} \mathcal{T}(\Sigma_{[m]}) \to \mathcal{T}(\Sigma_{[n]})$
delete components $n+1, \ldots, m$.

Lemma : projection

For all $n \leq m$, if $L \subseteq \mathcal{T}(\Sigma_{[m]})$ is regular then $proj_n(L)$ is regular.

cylindrification

$cyl_{n,m} : L \subseteq \mathcal{T}(\Sigma_{[n]}) \mapsto \{t \in \mathcal{T}(\Sigma_{[m]} \mid proj_n(t) \in L\}$ ($m \geq n$)

Lemma : cylindrification

For all $n \leq m$, if $L \subseteq \mathcal{T}(\Sigma_{[n]})$ is regular, then $cyl_{n,m}(L)$ est regular.

compilation : $X_1 \subseteq X_2$

Automate $[\![X_1 \subseteq X_2]\!]_2$:

- signature $\Sigma_{[2]} = \Sigma \times \{0,1\}^2$.
- states : q_0
- final states : q_0
- transitions :

$$\begin{array}{rcl} \langle a,0,0\rangle(q_0,\ldots,q_0) & \to & q_0 \\ \langle a,0,1\rangle(q_0,\ldots,q_0) & \to & q_0 \\ \langle a,1,1\rangle(q_0,\ldots,q_0) & \to & q_0 \end{array}$$

For $m \geq 2$,

$$[\![X_1 \subseteq X_2]\!]_m := cyl_{2,m}\big([\![X_1 \subseteq X_2]\!]_2\big)$$

compilation : $X_1 = X_2 \cdot 1$

Automate $[\![X_2 = X_1 \cdot 1]\!]_2$:

- signature $\Sigma_{[2]} = \Sigma \times \{0,1\}^2$.
- states : q_0, q_1, q_2
- final states : q_2
- transitions :

$$\begin{array}{lcl}
\langle a,0,0\rangle(q_0,\ldots,q_0) & \rightarrow & q_0 \\
\langle a,1,0\rangle(q_0,\ldots,q_0) & \rightarrow & q_1 \\
\langle a,0,1\rangle(q_1,q_0,\ldots,q_0) & \rightarrow & q_2 \\
\langle a,0,0\rangle(q_0,\ldots,q_0,q_2,q_0,\ldots,q_0) & \rightarrow & q_2
\end{array}$$

For $m \geq 2$,

$$[\![X_2 = X_1 \cdot 1]\!]_m := cyl_{2,m}\big([\![X_2 = X_1 \cdot 1]\!]_2\big)$$

compilation : $X_1 \subseteq L_a$

Automate $[\![X_1 \subseteq L_a]\!]_1$:

- signature $\Sigma_{[2]} = \Sigma \times \{0,1\}^2$.
- states : q_0
- final states : q_0
- transitions :

$$\begin{array}{rcll} \langle a,0\rangle(q_0,\ldots,q_0) & \to & q_0 & \\ \langle b,0\rangle(q_0,\ldots,q_0) & \to & q_0 & (b \neq a) \\ \langle a,1\rangle(q_0,\ldots,q_0) & \to & q_0 & \end{array}$$

For $m \geq 1$,

$$[\![X_1 \subseteq L_a]\!]_m := cyl_{1,m}\big([\![X_1 \subseteq L_a]\!]_1\big)$$

compilation : Boolean connectors

- $[\![\phi(X_1,\ldots,X_n) \vee \phi(X_1,\ldots,X_{n'})]\!]_m :=$
  $[\![\phi(X_1,\ldots,X_n)]\!]_m \cup [\![\phi(X_1,\ldots,X_{n'})]\!]_m$
  with $m \geq \max(n,n')$
- $[\![\phi(X_1,\ldots,X_n) \wedge \phi(X_1,\ldots,X_{n'})]\!]_m :=$
  $[\![\phi(X_1,\ldots,X_n)]\!]_m \cap [\![\phi(X_1,\ldots,X_{n'})]\!]_m$
  with $m \geq \max(n,n')$
- $[\![\neg\phi(X_1,\ldots,X_n)]\!]_m := \mathcal{T}(\Sigma_{[m]}) \setminus [\![\phi(X_1,\ldots,X_n)]\!]_m$
  for $m \geq n$.

compilation : quantifiers

- $[\![\exists X_{n+1}\, \phi(X_1, \ldots, X_{n+1})]\!]_n := \mathit{proj}_n\big([\![\phi(X_1, \ldots, X_{n+1})]\!]_{n+1}\big)$
- NB : this construction does not preserve determinism.
- $[\![\exists X_{n+1}\, \phi(X_1, \ldots, X_{n+1})]\!]_m :=$
  $\mathit{cyl}_{n,m}\big([\![\exists X_{n+1}\, \phi(X_1, \ldots, X_{n+1})]\!]_n\big)$ for $m \geq n$.
- $\forall = \neg\exists\neg$

Theorem Thatcher & Wright

Theorem :

For all formula $\phi \in \mathsf{WS}k\mathsf{S}_0$ over Σ without free variables, there exists a tree automaton $\mathcal{A}_\phi$ over Σ, such that $L(\mathcal{A}_\phi) = L(\phi)$.

$\mathcal{A}_\phi = [\![\phi]\!]_0$ can be computed explicitely !

Corollary :

For all formula $\phi \in \mathsf{WS}k\mathsf{S}$ over Σ without free variables there exists a tree automaton $\mathcal{A}_\phi$ over Σ, such that $L(\mathcal{A}_\phi) = L(\phi)$.

using translation of $\mathsf{WS}k\mathsf{S}$ into $\mathsf{WS}k\mathsf{S}_0$ first.

Size of $\mathcal{A}_\phi$

Theorem : Stockmeyer and Meyer 1973

For all n there exists $\exists x_1 \neg \exists y_1 \exists x_2 \neg \exists y_2 \ldots \exists x_n \neg \exists y_n \phi \in$ FOL such that for every automaton $\mathcal{A}$ recognizing the same language

$$\mathrm{size}(\mathcal{A}) \geq \left. 2^{2^{\cdots^{2^{\mathrm{size}(\phi)}}}} \right\} n$$

WSkS and FO

2 directions of the Thatcher & Wright theorem :

$$\text{WS}k\text{S} \ni \phi \mapsto \mathcal{A} \mapsto \exists Y_1 \ldots \exists Y_n \, \psi$$

with $\psi \in$ FOL.

Corollary :

Every WSkS formula is equivalent to a formula $\exists Y_1 \ldots \exists Y_n \, \psi$ with ψ first order.