Topics in Concurrency

Lecture 8

Jonathan Hayman

2 March 2015

Model checking modal-μ

Assume processes are finite-state

- Brute force (+ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]

Model checking modal-μ

Assume processes are finite-state

- Brute force (+ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]
  "Silly idea"

$$p \in \nu X.\varphi(X) \iff p \in \varphi(\nu X. \qquad \varphi(X))$$

Model checking modal-μ

Assume processes are finite-state

- Brute force ($+$ optimizations) computes each fixed point
- Local model checking [Larsen, Stirling and Walker, Winskel]
  Reduction Lemma

$$p \in \nu X.\varphi(X) \iff p \in \varphi(\nu X.\{p\} \vee \varphi(X))$$

Modal-μ for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

$$A ::= U \mid T \mid F \mid \neg A \mid A \wedge B \mid A \vee B \mid \langle a \rangle A \mid \langle - \rangle A \mid \nu X \qquad\qquad .A$$

Semantics identifies assertions with subsets of states:

- U is an arbitrary subset of states
- $T = \mathcal{S}$
- $F = \emptyset$
- $\neg A = \mathcal{S} \setminus A$
- $A \wedge B = A \cap B$
- $A \vee B = A \cup B$
- $\langle a \rangle A = \{p \in \mathcal{S} \mid \exists q. p \xrightarrow{a} q \wedge q \in A\}$
- $\langle - \rangle A = \{p \in \mathcal{S} \mid \exists q, a. p \xrightarrow{a} q \wedge q \in A\}$
- $\nu X\{p_1, \ldots, p_n\}.A = \bigcup\{U \subseteq S \mid U \subseteq \qquad\qquad A[U/X]\}$

Modal-μ for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

$$A ::= U \mid T \mid F \mid \neg A \mid A \wedge B \mid A \vee B \mid \langle a \rangle A \mid \langle - \rangle A \mid \nu X\{p_1, \ldots, p_n\}.A$$

Semantics identifies assertions with subsets of states:

- U is an arbitrary subset of states
- $T = \mathcal{S}$
- $F = \emptyset$
- $\neg A = \mathcal{S} \setminus A$
- $A \wedge B = A \cap B$
- $A \vee B = A \cup B$
- $\langle a \rangle A = \{p \in \mathcal{S} \mid \exists q. p \xrightarrow{a} q \wedge q \in A\}$
- $\langle - \rangle A = \{p \in \mathcal{S} \mid \exists q, a. p \xrightarrow{a} q \wedge q \in A\}$
- $\nu X\{p_1, \ldots, p_n\}.A = \bigcup\{U \subseteq S \mid U \subseteq \{p_1, \ldots, p_n\} \cup A[U/X]\}$

Modal-μ for model checking

Extend the syntax with defined basic assertions and adapt the fixed point operator:

$$A ::= U \mid T \mid F \mid \neg A \mid A \wedge B \mid A \vee B \mid \langle a \rangle A \mid \langle - \rangle A \mid \nu X\{p_1, \ldots, p_n\}.A$$

Semantics identifies assertions with subsets of states:

- U is an arbitrary subset of states
- $T = \mathcal{S}$
- $F = \emptyset$
- $\neg A = \mathcal{S} \setminus A$
- $A \wedge B = A \cap B$
- $A \vee B = A \cup B$
- $\langle a \rangle A = \{p \in \mathcal{S} \mid \exists q . p \xrightarrow{a} q \wedge q \in A\}$
- $\langle - \rangle A = \{p \in \mathcal{S} \mid \exists q, a . p \xrightarrow{a} q \wedge q \in A\}$
- $\nu X\{p_1, \ldots, p_n\}.A = \bigcup\{U \subseteq S \mid U \subseteq \{p_1, \ldots, p_n\} \cup A[U/X]\}$

As before, $\mu X.A \equiv \neg \nu X. \neg A[\neg X/X]$ and now

$$\nu X.A = \nu X\{\}.A$$

The reduction lemma

Lemma

Let $\varphi : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S})$ *be monotonic. For all* $U \subseteq \mathcal{S}$,

$$\begin{array}{rl} & U \subseteq \nu X.\varphi(X) \\ \iff & U \subseteq \varphi(\nu X.(U \cup \varphi(X))) \end{array}$$

In particular,

$$\begin{array}{rl} & p \in \nu X.\varphi(X) \\ \iff & p \in \varphi(\nu X.(\{p\} \cup \varphi(X))). \end{array}$$

Model checking algorithm

Given a transition system and a set of basic assertions $\{U, V, \ldots\}$:

$$
\begin{array}{llll}
p \vdash U & \longrightarrow & \text{true} & \text{if } p \in U \\
p \vdash U & \longrightarrow & \text{false} & \text{if } p \notin U \\
p \vdash T & \longrightarrow & \text{true} & \\
p \vdash F & \longrightarrow & \text{false} & \\
p \vdash \neg B & \longrightarrow & \text{not}(p \vdash B) & \\
p \vdash A \wedge B & \longrightarrow & p \vdash A \quad \text{and} \quad p \vdash B & \\
p \vdash A \vee B & \longrightarrow & p \vdash A \quad \text{or} \quad p \vdash B & \\
p \vdash \langle a \rangle B & \longrightarrow & q_1 \vdash B \text{ or } \ldots \text{ or } q_n \vdash B & \\
 & & \qquad \{q_1, \ldots, q_n\} = \{q \mid p \xrightarrow{a} q\} & \\
p \vdash \nu X\{\vec{r}\}.B & \longrightarrow & \text{true} & \text{if } p \in \{\vec{r}\} \\
p \vdash \nu X\{\vec{r}\}.B & \longrightarrow & p \vdash B[\nu X\{p, \vec{r}\}.B/X] & \text{if } p \notin \{\vec{r}\}
\end{array}
$$

Can use any sensible reduction technique for not, or and and.

Examples

Define the pure CCS process

$$P \stackrel{\mathrm{def}}{=} a.(a.\mathbf{nil} + a.P)$$

Check

$$P \vdash \nu X.\langle a \rangle X$$

and check

$$P \vdash \mu X.\langle a \rangle X$$

Note:

$$\mu Y.[-]F \vee \langle - \rangle Y \equiv \neg \nu Y.\neg([-]F \vee \langle - \rangle \neg Y))$$

Well-founded induction

A binary relation $\prec$ on a set A is well-founded iff there are no infinite descending chains

$$\cdots \prec a_n \prec \cdots \prec a_1 \prec a_0$$

The principle of well-founded induction:
Let $\prec$ be a well-founded relation on a set A. Let P be a property on A. Then

$$\forall a \in A.\ P(a)$$

iff

$$\forall a \in A.\ ((\forall b \prec a.\ P(b)) \implies P(a))$$

Correctness and termination of the algorithm

Write $(p \vDash A) = \mathrm{true}$ iff p is in the set of states determined by A.

Theorem

Let $p \in \mathcal{P}$ be a finite-state process and A be a closed assertion. For any truth value $t \in \{\mathrm{true}, \mathrm{false}\}$,

$$(p \vdash A) \to^{*} t \quad \Longleftrightarrow \quad (p \vDash A) = t$$

Proof sketch

For assertions A and A', take

$$A' \prec A \iff \begin{array}{ll} & A' \text{ is a proper subassertion of } A \\ \text{or} & A \equiv \nu X\{\vec{r}\}B \;\; \& \\ \exists p & A' \equiv \nu X\{\vec{r}, p\}B \;\; \& \;\; p \notin \vec{r} \end{array}$$

Want, for all closed assertions A,

$$Q(A) \quad \iff \quad \forall q \in \mathcal{P}.\forall t.(q \vdash A) \rightarrow^* t \iff (q \vDash A) = t$$

We show the following stronger property on open assertions by well-founded induction:

$$Q^+(A) \quad \iff \quad \begin{array}{l} \forall \text{closed substitutions for free variables} \\ B_1/X_1, \ldots, B_n/X_n : \\ Q(B_1) \& \ldots \& Q(B_n) \implies Q(A[B_1/X_1, \ldots, B_n/X_n]) \end{array}$$

The proof (presented in the lecture notes) centrally depends on the reduction lemma.

Chapter 6 Petri nets

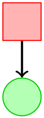
In interleaving models,

$$\alpha.\mathbf{nil} \parallel \beta.\mathbf{nil} \sim \alpha.\beta.\mathbf{nil} + \beta.\alpha.\mathbf{nil}$$

Petri nets:

Transitions Events

States Conditions



A wide range of applications:

- Fairness: In the following, does α ever occur?

- Partial order model checking
- Security models and event-based reasoning
- Hardware models
- Biology

∞-multisets

$$\omega^{\infty} = \omega \cup \{\infty\}$$

Extend addition:

$$n + \infty = \infty \qquad \text{for } n \in \omega^{\infty}$$

Extend subtraction

$$\infty - n = \infty \qquad \text{for } n \in \omega$$

Extend order:

$$n \leq \infty \qquad \text{for } n \in \omega^{\infty}$$

An ∞-multiset over a set X is a function

$$f : X \to \omega^{\infty}$$

It is a multiset if $f : X \to \omega$.

Operations on ∞-multisets

- $f \leq g$ iff $\forall x \in X. f(x) \leq g(x)$
- $f + g$ is the ∞-multiset such that

$$\forall x \in X.\ (f+g)(x) = f(x) + g(x)$$

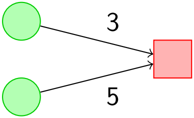
- For g a multiset such that $f \leq g$,

$$\forall x \in X.\ (f-g)(x) = f(x) - g(x)$$

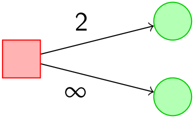
General Petri nets

A general Petri net consists of

- a set of conditions P 
- a set of events T 
- a pre-condition map assigning to each event t a multiset of conditions ${}^\bullet t$



- a post-condition map assigning to each event t an ∞-multiset of conditions $t^\bullet$



- a capacity map Cap an ∞-multiset of conditions, assigning a capacity in ω^∞ to each condition

Dynamics

A marking is an ∞-multiset $\mathcal{M}$ such that

$$\mathcal{M} \leq Cap$$

giving how many tokens are in each condition.

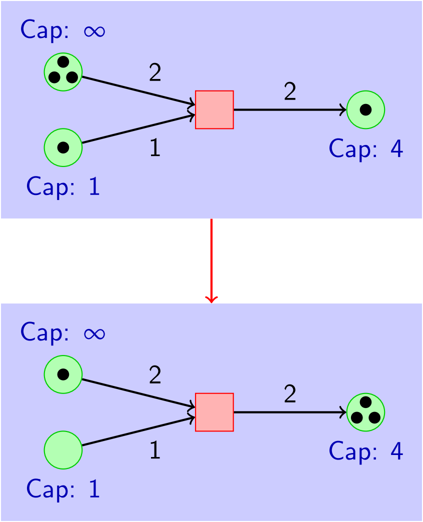
The token game:

For $\mathcal{M}, \mathcal{M}'$ markings, t an event:

$$\mathcal{M} \xrightarrow{t} \mathcal{M}' \quad \text{iff} \quad {}^{\bullet}t \leq \mathcal{M} \quad \& \quad \mathcal{M}' = \mathcal{M} - {}^{\bullet}t + t^{\bullet}$$

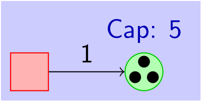
An event t has concession (is enabled) at $\mathcal{M}$ iff

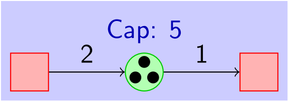
$${}^{\bullet}t \leq \mathcal{M} \quad \& \quad \mathcal{M} - {}^{\bullet}t + t^{\bullet} \leq Cap$$

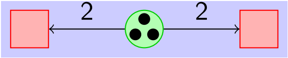


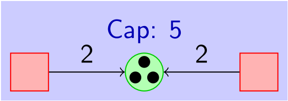
Cap: ∞
2
2
1
Cap: 4
Cap: 1
Cap: ∞
2
2
1
Cap: 4
Cap: 1

Further examples









Basic Petri nets

Often don't need multisets and can just consider sets.

A basic net consists of

- *a set of conditions* B
- *a set of events* E
- *a pre-condition map assigning a subset of conditions* ${}^\bullet e$ *to any event* e
- *a post-condition map assigning a subset of conditions* $e^\bullet$ *to any event* e *such that*

$$ {}^\bullet e \cup e^\bullet \neq \emptyset $$

The capacity of any condition is implicitly taken to be 1*:*

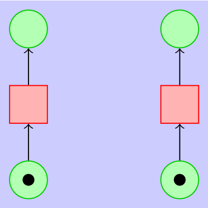
$$ \forall\, b \in B : \ Cap(b) = 1 $$

A marking $\mathcal{M}$ *is now a subset of conditions.*

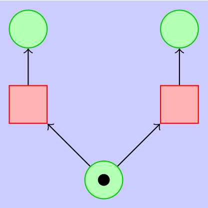
$$ \mathcal{M} \xrightarrow{e} \mathcal{M}' \qquad \text{iff} \qquad \begin{array}{ll} & {}^\bullet q \subseteq \mathcal{M} \quad \& \quad (\mathcal{M} \setminus {}^\bullet e) \cap e^\bullet = \emptyset \\ \& & \mathcal{M}' = (\mathcal{M} \setminus {}^\bullet e) \cup e^\bullet \end{array} $$

Concepts

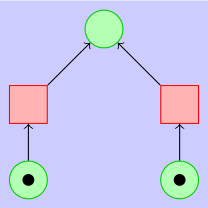
Concurrency



Forwards conflict



Backwards conflict



Contact