Topics in Concurrency

Lectures 6–7

Jonathan Hayman

27 October 2016

Specification logics

Logics for specifying correctness properties.
We'll look at:

- Basic logics and bisimilarity
- Fixed points and logic
- CTL
- Model checking

Finitary Hennessy-Milner Logic

Assertions:

$$A ::= T \mid F \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \neg A \mid \langle \lambda \rangle A \mid \langle - \rangle A \mid [\lambda] A \mid [-] A$$

Satisfaction: $s \vDash A$

Finitary Hennessy-Milner Logic

Assertions:

$$A ::= T \mid F \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \neg A \mid \langle\lambda\rangle A \mid \langle -\rangle A \mid [\lambda]A \mid [-]A$$

Satisfaction: $s \models A$

$$\begin{array}{rll}
s \models T & \text{always} & \\
s \models F & \text{never} & \\
s \models A_0 \wedge A_1 & \text{if} & s \models A_0 \quad \text{and} \quad s \models A_1 \\
s \models A_0 \vee A_1 & \text{if} & s \models A_0 \quad \text{or} \quad s \models A_1 \\
s \models \neg A & \text{if} & \text{not} \quad s \models A \\
s \models \langle\lambda\rangle A & \text{if} & \text{there exists } s' \text{ s.t. } s \xrightarrow{\lambda} s' \text{ and } s' \models A \\
s \models \langle -\rangle A & \text{if} & \text{there exist } s', \lambda \text{ s.t. } s \xrightarrow{\lambda} s' \text{ and } s' \models A \\
s \models [\lambda]A & \text{iff} & \text{for all } s' \text{ s.t. } s \xrightarrow{\lambda} s' \text{ have } s' \models A \\
s \models [-]A & \text{iff} & \text{for all } s', \lambda \text{ s.t. } s \xrightarrow{\lambda} s' \text{ have } s' \models A
\end{array}$$

Finitary Hennessy-Milner Logic

Assertions:

$$A ::= T \mid F \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \neg A \mid \langle\lambda\rangle A \mid \langle -\rangle A \mid [\lambda]A \mid [-]A$$

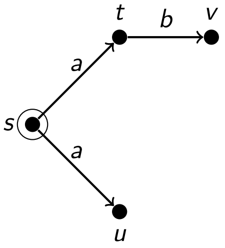
Satisfaction: $s \models A$

$$
\begin{array}{rll}
s \models T & \text{always} & \\
s \models F & \text{never} & \\
s \models A_0 \wedge A_1 & \text{if} & s \models A_0 \quad \text{and} \quad s \models A_1 \\
s \models A_0 \vee A_1 & \text{if} & s \models A_0 \quad \text{or} \quad s \models A_1 \\
s \models \neg A & \text{if} & \text{not} \quad s \models A \\
s \models \langle\lambda\rangle A & \text{if} & \text{there exists } s' \text{ s.t. } s \xrightarrow{\lambda} s' \text{ and } s' \models A \\
s \models \langle -\rangle A & \text{if} & \text{there exist } s', \lambda \text{ s.t. } s \xrightarrow{\lambda} s' \text{ and } s' \models A \\
s \models [\lambda]A & \text{iff} & \text{for all } s' \text{ s.t. } s \xrightarrow{\lambda} s' \text{ have } s' \models A \\
s \models [-]A & \text{iff} & \text{for all } s', \lambda \text{ s.t. } s \xrightarrow{\lambda} s' \text{ have } s' \models A
\end{array}
$$

Alternatively, derived assertions

$$[\lambda]A \equiv \neg\langle\lambda\rangle\neg A \qquad [-]A \equiv \neg\langle -\rangle\neg A$$

Examples



$? \; s \models \langle a \rangle T \; ?$

$? \; s \models [a] T \; ?$

$? \; u \models [-] F \; ?$

$? \; s \models \langle a \rangle \langle b \rangle T \; ?$

$? \; s \models [a] \langle b \rangle T \; ?$

Examples

Generally:

- $\langle a \rangle T$
- $[a] F$
- $\langle - \rangle F$
- $\langle - \rangle T$
- $[-] T$
- $[-] F$

(Strong) bisimilarity and logic

A non-finitary Hennessy-Milner logic allows an infinite conjunction

$$A ::= \bigwedge_{i \in I} A_i \mid \neg A \mid \langle \lambda \rangle A$$

with semantics

$$s \models \bigwedge_{i \in A} A_i \text{ iff } s \models A_i \text{ for all } i \in I$$

Define

$$p \asymp q \quad \text{iff} \quad \text{for all assertions } A \text{ of H-M logic}$$
$$p \models A \text{ iff } q \models A$$

Theorem

$$\asymp \quad = \quad \sim$$

This gives a way to demonstrate non-bisimilarity of states

Fixed points and model checking

- The finitary H-M logic doesn't allow properties such as

  the process never deadlocks

- We can add particular extensions (such as always, never) to the logic (CTL)
- Alternatively, what about defining sets of states 'recursively'? The set of states X that can always do some action satisfies:

$$X = \langle - \rangle T \wedge [-] X$$

Fixed points and model checking

- The finitary H-M logic doesn't allow properties such as

  the process never deadlocks

- We can add particular extensions (such as always, never) to the logic (CTL)
- Alternatively, what about defining sets of states 'recursively'? The set of states X that can always do some action satisfies:

$$X = \langle - \rangle T \wedge [-]X$$

- A fixed point equation: $X = \phi(X)$
- But such equations can have many solutions...

Fixed point equations

- In general, an equation of the form $X = \phi(X)$ can have many solutions for X.
- Fixed points are important: they represent steady or consistent states
- Range of different fixed point theorems applicable in different contexts e.g.

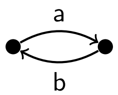
Theorem (1-dimensional Brouwer's fixed point theorem)

Any continuous function $f : [0,1] \to [0,1]$ *has at least one fixed point*

(used e.g. in proof of existence of Nash equilibria)

- We'll be interested in fixed points of functions on the powerset lattice $\rightsquigarrow$ Knaster-Tarski fixed point theorem and least and greatest fixed points

Least and greatest fixed points on transition systems: examples



In the above transition system, what are the least and greatest subsets of states X, Y and Z that satisfy:

$$X = X$$

$$Y = \langle - \rangle T \wedge [-] Y$$

$$Z = \neg Z$$

The powerset lattice

- Given a set $\mathcal{S}$, its powerset is

$$\mathcal{P}(\mathcal{S}) = \{S \mid S \subseteq \mathcal{S}\}$$

- Taking the order on its elements to be inclusion, $\subseteq$, this forms a complete lattice

The powerset lattice

- Given a set $\mathcal{S}$, its powerset is

$$\mathcal{P}(\mathcal{S}) = \{S \mid S \subseteq \mathcal{S}\}$$

- Taking the order on its elements to be inclusion, $\subseteq$, this forms a complete lattice

We are interested in fixed points of functions of the form

$$\phi : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S})$$

- ϕ is monotonic if $S \subseteq S'$ implies $\phi(S) \subseteq \phi(S')$
- a prefixed point of ϕ is a set X satisfying $\phi(X) \subseteq X$
- a postfixed point of ϕ is a set X satisfying $X \subseteq \phi(X)$

Knaster-Tarski fixed point theorem for minimum fixed points

Theorem

For monotonic $\phi : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S})$, define

$$m = \bigcap \{ X \subseteq \mathcal{S} \mid \phi(X) \subseteq X \}.$$

Then m is a fixed point of ϕ and, furthermore, is the least prefixed point:

1. $m = \phi(m)$
2. $\phi(X) \subseteq X$ *implies* $m \subseteq X$

m is conventionally written

$$\color{red}{\mu X.\phi(X)}$$

Used for inductive definitions: syntax, operational semantics, rule-based programs, model checking

Knaster-Tarski fixed point theorem for maximum fixed points

Theorem

For monotonic $\phi : \mathcal{P}(\mathcal{S}) \to \mathcal{P}(\mathcal{S})$, define

$$M = \bigcup \{X \subseteq \mathcal{S} \mid X \subseteq \phi(X)\}.$$

Then M is a fixed point of ϕ and, furthermore, is the greatest postfixed point.

1. $M = \phi(M)$
2. $X \subseteq \phi(X)$ *implies* $X \subseteq M$

M is conventionally written

$$\nu X.\phi(X)$$

Used for co-inductive definitions, bisimulation, model checking

(Strong) bisimilarity as a maximum fixed point [§5.2 p68]

Bisimilarity can be viewed as a fixed point $\rightsquigarrow$ model checking algorithms.

Given a relation R (on CCS processes or states of transition systems) define:

$$p \; \phi(R) \; q$$

iff

1. $\forall \alpha, p'. \quad p \xrightarrow{\alpha} p' \quad \Longrightarrow$
   $\exists q'. \quad q \xrightarrow{\alpha} q' \quad \& \quad p' \; R \; q'$
2. $\forall \alpha, q'. \quad q \xrightarrow{\alpha} q' \quad \Longrightarrow$
   $\exists p'. \quad p \xrightarrow{\alpha} p' \quad \& \quad p' \; R \; q'$

Lemma

$R \subseteq \phi(R)$ iff R is a (strong) bisimulation.

Hence, by Knaster-Tarski fixed point theorem for maximum fixed points:

Theorem

Bisimilarity is the greatest fixed point of ϕ.

Theorem

Bisimilarity is the greatest fixed point of ϕ.

Proof.

$$\begin{aligned} \sim \quad &= \quad \bigcup\{R \mid R \text{ is a bisimulation}\} && (1) \\ &= \quad \bigcup\{R \mid R \subseteq \phi(R)\} && (2) \\ &= \quad \nu X.\phi(X) && (3) \end{aligned}$$

(1) is by definition of $\sim$
(2) is by Lemma
(3) is by Knaster-Tarski for maximum fixed points: note that ϕ is monotonic

$\square$

Theorem

Bisimilarity is the greatest fixed point of ϕ.

Proof.

$$
\begin{aligned}
\sim \quad &= \quad \bigcup\{R \mid R \text{ is a bisimulation}\} && (1)\\
&= \quad \bigcup\{R \mid R \subseteq \phi(R)\} && (2)\\
&= \quad \nu X.\phi(X) && (3)
\end{aligned}
$$

(1) is by definition of $\sim$
(2) is by Lemma
(3) is by Knaster-Tarski for maximum fixed points: note that ϕ is monotonic

$\square$

Question: How is this different from the least fixed point of ϕ?