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:

\[ A ::= T | F | A_0 \land A_1 | A_0 \lor A_1 | \neg A | \langle \lambda \rangle A | \langle - \rangle A | [\lambda] A | [-] A \]

Satisfaction: \( s \models A \)
Finitary Hennessy-Milner Logic

Assertions:

\[ A ::= T \mid F \mid A_0 \land A_1 \mid A_0 \lor A_1 \mid \neg A \mid \langle \lambda \rangle A \mid \langle - \rangle A \mid [\lambda]A \mid [-]A \]

Satisfaction: \( s \models A \)

- \( s \models T \) always
- \( s \models F \) never
- \( s \models A_0 \land A_1 \) if \( s \models A_0 \) and \( s \models A_1 \)
- \( s \models A_0 \lor A_1 \) if \( s \models A_0 \) or \( s \models A_1 \)
- \( s \models \neg A \) if not \( s \models A \)
- \( s \models \langle \lambda \rangle A \) if there exists \( s' \) s.t. \( s \xrightarrow{\lambda} s' \) and \( s' \models A \)
- \( s \models \langle - \rangle A \) if there exist \( s', \lambda \) s.t. \( s \xrightarrow{\lambda} s' \) and \( s' \models A \)
- \( s \models [\lambda]A \) iff for all \( s' \) s.t. \( s \xrightarrow{\lambda} s' \) have \( s' \models A \)
- \( s \models [-]A \) iff for all \( s', \lambda \) s.t. \( s \xrightarrow{\lambda} s' \) have \( s' \models A \)

Alternatively, derived assertions

\[ [\lambda]A \equiv \neg [\neg]A \]

\[ [-]A \equiv \neg [-]A \]
Finitary Hennessy-Milner Logic

Assertions:

\[ A ::= \mathit{T} \mid \mathit{F} \mid A_0 \land A_1 \mid A_0 \lor A_1 \mid \neg A \mid \langle \lambda \rangle A \mid \langle \neg \rangle A \mid [\lambda]A \mid [\neg]A \]

Satisfaction: \( s \models A \)

\[ s \models T \quad \text{always} \]
\[ s \models F \quad \text{never} \]
\[ s \models A_0 \land A_1 \quad \text{if} \quad s \models A_0 \quad \text{and} \quad s \models A_1 \]
\[ s \models A_0 \lor A_1 \quad \text{if} \quad s \models A_0 \quad \text{or} \quad s \models A_1 \]
\[ s \models \neg A \quad \text{if} \quad \text{not} \quad s \models A \]
\[ s \models \langle \lambda \rangle A \quad \text{if} \quad \text{there exists} \ s' \ \text{s.t.} \ s \xrightarrow{\lambda} s' \ \text{and} \ s' \models A \]
\[ s \models \langle \neg \rangle A \quad \text{if} \quad \text{there exist} \ s', \lambda \ \text{s.t.} \ s \xrightarrow{\lambda} s' \ \text{and} \ s' \models A \]
\[ s \models [\lambda]A \quad \text{iff} \quad \text{for all} \ s' \ \text{s.t.} \ s \xrightarrow{\lambda} s' \ \text{have} \ s' \models A \]
\[ s \models [\neg]A \quad \text{iff} \quad \text{for all} \ s', \lambda \ \text{s.t.} \ s \xrightarrow{\lambda} s' \ \text{have} \ s' \models A \]

Alternatively, derived assertions

\[ [\lambda]A \equiv \neg \langle \lambda \rangle \neg A \quad [\neg]A \equiv \neg \langle \neg \rangle \neg A \]
Examples

\[ \mathcal{S} = \langle a \rangle T \]
\[ \mathcal{S} = [a] T \]
\[ \mathcal{U} = [-] F \]
\[ \mathcal{S} = \langle a \rangle \langle b \rangle T \]
\[ \mathcal{S} = [a] \langle b \rangle T \]
Examples

Generally:

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

\[ s \models \bigwedge_{i \in I} A_i \text{ iff } s \models A_i \text{ for all } i \in I \]

Define

\[ p \bowtie q \text{ iff for all assertions } A \text{ of H-M logic } \]

\[ p \models A \text{ iff } q \models A \]

**Theorem**

\[ \bowtie = \sim \]

This gives a way to demonstrate non-bisimilarity of states
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:

$$X = (\neg) T \land [\neg] X$$
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:

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

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 $\mathcal{P}$ 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:

\[
X = X
\]

\[
Y = (\neg T \land [\neg] Y
\]

\[
Z = \neg Z
\]
The powerset lattice

• Given a set $S$, its powerset is

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

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

- Given a set $S$, its powerset is

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

- 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

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

- $\phi$ is monotonic if $S \subseteq S'$ implies $\phi(S) \subseteq \phi(S')$
- A prefixed point of $\phi$ is a set $X$ satisfying $\phi(X) \subseteq X$
- A postfixed point of $\phi$ is a set $X$ satisfying $X \subseteq \phi(X)$
Knaster-Tarski fixed point theorem for minimum fixed points

Theorem

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

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

Then $m$ is a fixed point of $\phi$ and, furthermore, is the least prefixed point:

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

$m$ is conventionally written

$$\mu X.\phi(X)$$

Used for inductive definitions: syntax, operational semantics, rule-based programs, model checking
Knaster-Tarski fixed point theorem for maximum fixed points

**Theorem**

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

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

Then $M$ is a fixed point of $\phi$ 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
Bisimilarity can be viewed as a fixed point $\sim$ model checking algorithms.

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

$$p \phi(R) q$$

iff

1. $\forall \alpha, p'. \ p \xrightarrow{\alpha} p' \implies \exists q'. \ q \xrightarrow{\alpha} q' \& p' R q'$

2. $\forall \alpha, q'. \ q \xrightarrow{\alpha} q' \implies \exists p'. \ p \xrightarrow{\alpha} p' \& p' R q'$

**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 $\phi$. 
Theorem

Bisimilarity is the greatest fixed point of $\phi$.

Proof.

\[
\sim = \bigcup \{ R \mid R \text{ is a bisimulation} \} \tag{1}
\]
\[
= \bigcup \{ R \mid R \subseteq \phi(R) \} \tag{2}
\]
\[
= \nu X.\phi(X) \tag{3}
\]

(1) is by definition of $\sim$

(2) is by Lemma

(3) is by Knaster-Tarski for maximum fixed points: note that $\phi$ is monotonic
**Theorem**

*Bisimilarity is the greatest fixed point of \( \phi \).*

**Proof.**

\[
\sim = \bigcup \{ R \mid R \text{ is a bisimulation} \} \tag{1}
\]
\[
= \bigcup \{ R \mid R \subseteq \phi(R) \} \tag{2}
\]
\[
= \nu X.\phi(X) \tag{3}
\]

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

**Question:** How is this different from the least fixed point of \( \phi \)?