Chapter 6. Hennessy-Milner Logic with recursive definitions


Hernán Vanzetto

INRIA Nancy & LORIA, France

VINO 2011, Campo Tures, Italy

July 16, 2011
Syntactic

For an action $a \in Act$

$$F ::= tt \mid ff \mid F \land F \mid F \lor F \mid \langle a \rangle F \mid [a]F$$

where

- $\langle a \rangle F$ states that it is possible to perform action $a$ and thereby (in the next state) satisfy $F$
- $[a]F$ states that no matter how a process performs action $a$, the state it reaches afterwards necessarily satisfy $F$
Hennessy-Milner Logic

Semantics

For each formula $F$, associate a set of states where the formula is valid. $\llbracket F \rrbracket \subseteq \text{Proc}$ is defined inductively by

1. $\llbracket \mathsf{tt} \rrbracket = \text{Proc}$
2. $\llbracket \mathsf{ff} \rrbracket = \emptyset$
3. $\llbracket F \land G \rrbracket = \llbracket F \rrbracket \cap \llbracket G \rrbracket$
4. $\llbracket F \lor G \rrbracket = \llbracket F \rrbracket \cup \llbracket G \rrbracket$
5. $\llbracket \langle a \rangle F \rrbracket = \langle \cdot a \cdot \rangle \llbracket F \rrbracket$
6. $\llbracket [a]F \rrbracket = \langle a \cdot \rangle \llbracket F \rrbracket$

where $\langle \cdot a \cdot \rangle$, $\langle a \cdot \rangle : 2^{\text{Proc}} \to 2^{\text{Proc}}$ are defined by

- $\langle \cdot a \cdot \rangle S = \{ p \in \text{Proc} \mid \exists p'. p \xrightarrow{a} p' \land p' \in S \}$
- $\langle a \cdot \rangle S = \{ p \in \text{Proc} \mid \forall p'. p \xrightarrow{a} p' \Rightarrow p' \in S \}$
Hennessy-Milner Logic

Temporal properties not expressible in HML

- Inv(F) iff all reachable states satisfy F
  \[
  Inv(F) = F \land [Act]F \land [Act][Act]F \land [Act][Act][Act]F \land \ldots \\
  = \bigwedge_{i \geq 0} [a]^i F
  \]

- Pos(F) iff there is a reachable state which satisfies F
  \[
  Pos(F) = F \lor \langle Act \rangle F \lor \langle Act \rangle \langle Act \rangle F \lor \langle Act \rangle \langle Act \rangle \langle Act \rangle F \lor \ldots \\
  = \bigvee_{i \geq 0} \langle a \rangle^i F
  \]

Problems

- infinite formulae cannot be expressed in HML
- infinite formulae are difficult to handle
Syntax

Formulae are given by the following abstract syntax

\[ F ::= X \mid tt \mid ff \mid F \land F \mid F \lor F \mid \langle a \rangle F \mid [a]F \]

where \( a \in Act \), and \( X \) is a distinguished variable with a definition

- \( X^{\min} \equiv F_X \), or \( X^{\max} \equiv F_X \)

such that \( F_X \) is a formula of the logic which can contain \( X \).
Example:

\[ X^{\min} = X \]

Any set of states \( S \) satisfies the set-equation \( X = X \). The least such set is \( \emptyset \).

Example:

\[ X^{\max} = X \]

Any set of states \( S \) satisfies the set-equation \( X = X \). The greatest such set is \( Proc \).
HML with one recursively defined variable

Example: “A state can be reached where $a$ cannot be executed”

$$X \leftarrow [a]ff \lor \langle Act \rangle X$$

The property is valid for the labelled transition system. The sets $\{1, 2\}$ and $\{0, 1, 2\}$ are solutions for the equation.

But what we intended to describe is the least solution!

$$X \overset{\text{min}}{=} [a]ff \lor \langle Act \rangle X$$
Example: “A state can be reached where $a$ cannot be executed”

\[ X \overset{\text{min}}{=} [a] ff \lor \langle Act \rangle X \]

The unique least solution for this equation is the set of states $\emptyset$. Hence the property is not valid for this labeled transition system.
HML with one recursively defined variable

Example: “In every reachable state an a-transition is possible”

\[ X \equiv \langle a \rangle tt \land [\text{Act}]X \]

Solutions: \( \emptyset \), \{0\} and \{0, 1\}

What we intended to describe is the greatest solution!

\[ X^{\text{max}} \equiv \langle a \rangle tt \lor [\text{Act}]X \]
HML with one recursively defined variable

Example: “In every reachable state an \( a \)-transition is possible”

\[
X \overset{\text{max}}{=} \langle a \rangle \top \land [\text{Act}]X
\]

The greatest solution for this equation is the set of states \( \{0\} \).
Thus property is not valid for the labeled transition system.
Example: “In all states reachable by a $b$-transition (0 or more), a 
$b$-transition is possible”

$$X \overset{\text{max}}{=} \langle b \rangle tt \land [b]X$$

The greatest solution is $\{s_1, s_2, t_1\}$. 
Formulas for the properties that cannot be expressed in HML:

- “the computer scientists never drinks coffee”
  \[ X^{\text{max}} = \neg \left[ \text{coffee} \right] \land [\text{Act}]X \]

- “the computer scientists always produces a publication after drinking wine”
  \[ X^{\text{max}} = \left[ \text{wine} \right] \left( \langle \text{pub} \rangle tt \land \neg [\text{Act}\\{\text{pub}\}] \right) \land [\text{Act}]X \]

- Inv(F)
  \[ X^{\text{max}} = F \land [\text{Act}]X \]

- Pos(F)
  \[ X^{\text{min}} = F \lor \langle \text{Act} \rangle X \]

If there’s more than one, which solution to choose? In general:

- min are used to express that something will happen sooner or later.
- max are used to express the invariance of some property during an execution or that something does not happen.
HML with one recursively defined variable

Semantics

- With each formula $F$ associate a set of states for which
  \[ [F] \subseteq Proc \]
  is satisfied.

- How to deal with recursion variable $X$?
  Make an assumption on states satisfied by $X$.

For every formula $F$ we define a function $O_F : 2^{Proc} \rightarrow 2^{Proc}$ s.t.
- if $S$ is the set of processes that satisfy $X$
- then $O_F(S)$ is the set of processes that satisfy $F$. 
HML with one recursively defined variable. Semantics

Definition of $\mathcal{O}_F : 2^{\text{Proc}} \rightarrow 2^{\text{Proc}}$

For $S \subseteq \text{Proc}$:

- $\mathcal{O}_\times(S) = S$
- $\mathcal{O}_{tt}(S) = \text{Proc}$
- $\mathcal{O}_{ff}(S) = \emptyset$
- $\mathcal{O}_{F_1 \land F_2}(S) = \mathcal{O}_{F_1}(S) \cap \mathcal{O}_{F_2}(S)$
- $\mathcal{O}_{F_1 \lor F_2}(S) = \mathcal{O}_{F_1}(S) \cup \mathcal{O}_{F_2}(S)$
- $\mathcal{O}_{\langle a \rangle F}(S) = \langle a \rangle \mathcal{O}_F S$
- $\mathcal{O}_{[a] F}(S) = [a] \mathcal{O}_F S$
HML with one recursively defined variable. Semantics

\[ \mathcal{O}_{\langle a \rangle} X(\{s\}) = \langle \cdot a \cdot \rangle \mathcal{O} X(\{s\}) = \langle \cdot a \cdot \rangle \{s\} = \{s_2\} \]

\[ \mathcal{O}_{\langle a \rangle} X(\{s, s_1\}) = \langle \cdot a \cdot \rangle \mathcal{O} X(\{s, s_1\}) = \langle \cdot a \cdot \rangle \{s, s_1\} = \{s, s_2\} \]

\[ \mathcal{O}_{[b]} X(\{s_1\}) = [\cdot b \cdot] \mathcal{O} X(\{s_1\}) = [\cdot b \cdot] \{s_1\} = \{s_1, s_2\} \]
We know that \((2^{\text{Proc}}, \subseteq)\) is a complete lattice and \(\mathcal{O}_F\) is monotonic, so \(\mathcal{O}_F\) has a unique least fixed point and a unique greatest fixed point (by Tarski's Fixed Point Theorem).

**Semantics of formula \(F\)**

1. \([tt] = \text{Proc}\)
2. \([ff] = \emptyset\)
3. \([F \land G] = [F] \cap [G]\)
4. \([F \lor G] = [F] \cup [G]\)
5. \([\langle a \rangle F] = \langle \cdot a \cdot \rangle [F]\)
6. \([\lbrack a \rbrack F] = [\cdot a \cdot][F]\)
7. If \(X \equiv F_X\) then \([X] = \bigcap\{S \subseteq \text{Proc} \mid S = \mathcal{O}_{F_X}(S)\}\)
8. If \(X \equiv F_X\) then \([X] = \bigcup\{S \subseteq \text{Proc} \mid S = \mathcal{O}_{F_X}(S)\}\)
HML with one recursively defined variable.

Notation

\[ T, s \models F \quad \text{iff} \quad s \in [F] \]

Let \( \text{Proc} \) be a finite set.

Computing the solution of \( X^{\min} = F_X \)

There exists a natural number \( m > 0 \) such that \([X] = \mathcal{O}_{F_X}^m(\emptyset)\)

Computing the solution of \( X^{\max} = F_X \)

There exist a natural number \( M > 0 \) such that \([X] = \mathcal{O}_{F_X}^M(\text{Proc})\)
HML with one recursively defined variable

Example: \( X \overset{\text{min}}{=} [a]ff \lor \langle \text{Act} \rangle X \)

\[
\begin{align*}
\mathcal{O}_{F_X}(S) & = \mathcal{O}_{[a]ff}(S) \cup \mathcal{O}_{\langle \text{Act} \rangle X}(S) \\
& = \{ \cdot a \cdot \} \mathcal{O}_{ff}(S) \cup \langle \cdot \text{Act} \cdot \rangle \mathcal{O}_X(S) \\
& = \{ \cdot a \cdot \} \emptyset \cup \langle \cdot \text{Act} \cdot \rangle S \\
& = \{2\} \cup \langle \cdot \text{Act} \cdot \rangle S
\end{align*}
\]

1. \( \mathcal{O}_{F_X}(\emptyset) = \{2\} \cup \langle \cdot \text{Act} \cdot \rangle \emptyset = \{2\} \cup \emptyset = \{2\} \)
2. \( \mathcal{O}_{F_X}(\{2\}) = \{2\} \cup \langle \cdot \text{Act} \cdot \rangle \{2\} = \{2\} \cup \{0\} = \{0, 2\} \)
3. \( \mathcal{O}_{F_X}(\{0, 2\}) = \{2\} \cup \langle \cdot \text{Act} \cdot \rangle \{0, 2\} = \{2\} \cup \{0\} = \{0, 2\} \)
HML with one recursively defined variable

Example: \( X^{\text{max}} = \langle b \rangle tt \land [b]X \)

\[
\begin{align*}
\mathcal{O}_{F_X}(S) &= \mathcal{O}_{\langle b \rangle tt}(S) \cap \mathcal{O}_{[b]X}(S) \\
&= \langle \cdot b \cdot \rangle \mathcal{O}_{tt}(S) \cap [\cdot b \cdot] \mathcal{O}_{X}(S) \\
&= \langle \cdot b \cdot \rangle \text{Proc} \cap [\cdot b \cdot]S \\
&= \{s_1, s_2, t_1\} \cap [\cdot b \cdot]S
\end{align*}
\]

1. \( \mathcal{O}_{F_X}(\text{Proc}) = \{s_1, s_2, t_1\} \cap [\cdot b \cdot]\text{Proc} = \{s_1, s_2, t_1\} \cap \{s, s_1, s_2, t, t_1\} = \{s_1, s_2, t_1\} \)
2. \( \mathcal{O}_{F_X}(\{s_1, s_2, t_1\}) = \{s_1, s_2, t_1\} \cap [\cdot b \cdot]\{s_1, s_2, t_1\} = \{s_1, s_2, t_1\} \cap \{s, s_1, s_2, t, t_1\} = \{s_1, s_2, t_1\} \)
Some temporal properties. Safety and liveness

- **Safe**($F$): formula $F$ holds in all the transition sequence
  \[ X^{\text{max}} = F \land ([\text{Act}]ff \lor \langle \text{Act} \rangle X) \]

- **Even**($F$): eventually $F$ will hold (in every execution)
  \[ X^{\text{min}} = F \lor (\langle \text{Act} \rangle tt \land [\text{Act}]X) \]

- $F \cup^w G$ (weak until): $F$ holds in all states until a state is reached where $G$ holds (but maybe this will never happen!)
  \[ X^{\text{max}} = G \lor (F \land [\text{Act}]X) \]

- $F \cup^s G$ (strong until): sooner or later $G$ holds and, until then, $F$ holds in all states traversed
  \[ X^{\text{min}} = G \lor (F \land \langle \text{Act} \rangle tt \land [\text{Act}]X) \]

- Even($F$) $\equiv$ tt $\cup^s G$

- Inv($F$) $\equiv$ $F \cup^w ff$

Duality:
- $\neg$Inv($F$) $\equiv$ Pos($\neg F$)
- $\neg$Safe($F$) $\equiv$ Even($\neg F$)
Mutually recursive equational systems

Multiple recursion variables: $X_1, X_2, \ldots, X_n$

\[
\begin{align*}
X_1 & \overset{\text{min/max}}{=} F_1(X_1, X_2, \ldots, X_n) \\
X_2 & \overset{\text{min/max}}{=} F_2(X_1, X_2, \ldots, X_n) \\
& \vdots \\
X_m & \overset{\text{min/max}}{=} F_m(X_1, X_2, \ldots, X_n)
\end{align*}
\]

where all equations are either lfp or all are gfp
and each $F_i(X_1, X_2, \ldots, X_n)$ is a formula generated by

\[
F ::= tt \mid ff \mid F \land F \mid F \lor F \mid \langle a \rangle F \mid [a]F \mid X_1 \mid X_2 \mid \ldots \mid X_n
\]
Mutually recursive equational systems

- A variable may depend on another variable

  “It is impossible to do two consecutive coin actions without a tea action in between”

  \[ X^{\text{max}} = [\text{coin}] Y \land [tt] X \]
  \[ Y^{\text{max}} = [\text{coin}] ff \land [\text{tea}] Y \]

- Variables may even depend on each other cyclicly

  \[ X^{\text{max}} = [a] Y \]
  \[ Y^{\text{max}} = \langle a \rangle X \]
Mutually recursive equational systems

Semantics of an equational systems

\[
\begin{align*}
X_1^{\min/\max} &= F_1(X_1, X_2, \ldots, X_n) \\
X_2^{\min/\max} &= F_2(X_1, X_2, \ldots, X_n) \\
&\vdots \\
X_m^{\min/\max} &= F_m(X_1, X_2, \ldots, X_n)
\end{align*}
\]

where all equations are either lfp or all are gfp.

- The semantics associates a set of states to each recursion variable
- The new domain is \( \mathcal{D} = (2^{Proc})^n \)
The pair \((\mathcal{D}, \sqsubseteq)\) with\[
(S_1, \ldots, S_n) \sqsubseteq (S'_1, \ldots, S'_n) \text{ iff } S_1 \subseteq S'_1 \land \ldots \land S_n \subseteq S'_n
\]
is a complete lattice.

**Notation:**
\[
(S_1, \ldots, S_n) \cap (S'_1, \ldots, S'_n)' = (S_1 \cap S'_1, \ldots, S_n \cap S'_n)
\]
\[
(S_1, \ldots, S_n) \cup (S'_1, \ldots, S'_n)' = (S_1 \cup S'_1, \ldots, S_n \cup S'_n)
\]
Mutually recursive equational systems

- Define $O_F : \mathcal{D} \Rightarrow 2^{\text{Proc}}$ such that $O_F(S_1, \ldots, S_n)$ is the set of states for which formula $F$ holds under the assumption that $X_i$ holds precisely in the states from $S_i$.

\[
\begin{align*}
O_{X_i}(S_1, \ldots, S_n) &= S_i \\
O_{tt}(S_1, \ldots, S_n) &= \text{Proc} \\
O_{ff}(S_1, \ldots, S_n) &= \emptyset \\
O_{F_1 \land F_2}(S_1, \ldots, S_n) &= O_{F_1}(S_1, \ldots, S_n) \cap O_{F_2}(S_1, \ldots, S_n) \\
O_{F_1 \lor F_2}(S_1, \ldots, S_n) &= O_{F_1}(S_1, \ldots, S_n) \cup O_{F_2}(S_1, \ldots, S_n) \\
O_{\langle a \rangle F}(S_1, \ldots, S_n) &= \langle \cdot a \cdot \rangle O_F(S_1, \ldots, S_n) \\
O_{[a]F}(S_1, \ldots, S_n) &= [\cdot a \cdot] O_F(S_1, \ldots, S_n)
\end{align*}
\]
Mutually recursive equational systems

- Define $\llbracket D \rrbracket = D \rightarrow D$
  $$\llbracket D \rrbracket(S_1, \ldots, S_n) = (\mathcal{O}_{F_1}(S_1, \ldots, S_n)), \ldots, \mathcal{O}_{F_n}(S_1, \ldots, S_n))$$
- $\llbracket D \rrbracket$ is monotonic over $(D, \sqsubseteq)$
- Define $\llbracket X_1, \ldots, X_n \rrbracket \in D$
  For a system of least fixed point equations take the least fixed point of $\llbracket D \rrbracket$:
  $$\llbracket X_1, \ldots, X_n \rrbracket = \sqcap\{(S_1, \ldots, S_n) \in D \mid \llbracket D \rrbracket(S_1, \ldots, S_n) = (S_1, \ldots, S_n)\}$$
  For a system of greatest fixed point equations take the greatest fixed point of $\llbracket D \rrbracket$:
  $$\llbracket X_1, \ldots, X_n \rrbracket = \sqcup\{(S_1, \ldots, S_n) \in D \mid \llbracket D \rrbracket(S_1, \ldots, S_n) = (S_1, \ldots, S_n)\}$$
Mutually recursive equational systems

Computing the solution of a block of least fixed point equations

Let $Proc$ be a finite set.

Let $m > 0$ such that

$$\begin{align*}
X_1 &\overset{\text{min}}{=} F_1(X_1, X_2, \ldots, X_n) \\
X_2 &\overset{\text{min}}{=} F_2(X_1, X_2, \ldots, X_n) \\
\vdots \\
X_m &\overset{\text{min}}{=} F_m(X_1, X_2, \ldots, X_n)
\end{align*}$$

There exists a natural number $m > 0$ such that

$$\llbracket X_1, \ldots, X_n \rrbracket = \llbracket D \rrbracket^m(\emptyset, \ldots, \emptyset)$$
Mutually recursive equational systems

Computing the solution of a block of greatest fixed point equations

Let $Proc$ be a finite set.

\[
\begin{align*}
X_1 \overset{\text{max}}{=} & \quad F_1(X_1, X_2, \ldots, X_n) \\
X_2 \overset{\text{max}}{=} & \quad F_2(X_1, X_2, \ldots, X_n) \\
\vdots \\
X_m \overset{\text{max}}{=} & \quad F_m(X_1, X_2, \ldots, X_n)
\end{align*}
\]

There exists a natural number $M > 0$ such that

\[
\lbrack X_1, \ldots, X_n \rbrack = \lbrack D \rbrack^M(Proc, \ldots, Proc)
\]
Example: Consider

\[ X^{\text{max}} = \langle a \rangle Y \land [a]Y \land [b]ff \]
\[ Y^{\text{max}} = \langle b \rangle x \land [b]X \land [a]ff \]

\[
\begin{pmatrix}
S_1 \\
S_2
\end{pmatrix}
\mapsto
\begin{pmatrix}
\langle \cdot a \cdot \rangle S_2 \cap [\cdot a \cdot]S_2 \cap \{s_1, s_3\}
\\
\langle \cdot b \cdot \rangle S_1 \cap [\cdot b \cdot]S_1 \cap \{s_2, s_4\}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\text{Proc} \\
\text{Proc}
\end{pmatrix}
\leadsto
\begin{pmatrix}
\{s_1, s_3\}
\\
\{s_2, s_4\}
\end{pmatrix}
\leadsto
\begin{pmatrix}
\{s_1, s_3\}
\\
\{s_2\}
\end{pmatrix}
\leadsto
\begin{pmatrix}
\{s_1\}
\\
\{s_2\}
\end{pmatrix}
\leadsto
\begin{pmatrix}
\{s_1\}
\\
\{s_2\}
\end{pmatrix}
\]
Characteristic properties

- Given a finite transition system \( p \),
  we want to find a formula \( F_p \) such that for all processes \( q \)

\[
q \models F_p \iff q \sim p
\]

- For image-finite processes, the equivalent class that contains \( p \) is
  \([p]_\sim = \{ q \mid q \sim p \}\).

**Theorem**

If the LTS is finite, then we can characterize the equivalence classes for strong bisimulation with a single formula and the formula is unique.

This formula is called the characteristic formula.
Characteristic properties

Example:

\[
\begin{array}{c}
p \quad \text{to} \quad q \\
\hline
m \quad \text{to} \quad m \\
\hline
\bar{t} \quad \text{to} \quad \bar{t} \\
\hline
\bar{k} \quad \text{to} \quad \bar{k}
\end{array}
\]

\(p\) characteristic formula can be constructed as follows:

- \(p\) can perform \(m\) and become \(q\).
- No matter how \(p\) performs \(m\) it becomes \(q\).
- \(p\) cannot perform any action other than \(m\).

\[X_p = \langle m \rangle X_q \land [m]X_q \land [\{\bar{t}, \bar{k}\}] ff\]

\(q\) characteristic formula is:

\[X_q = \langle \bar{t} \rangle X_p \land \langle \bar{k} \rangle X_p \land [\{\bar{t}, \bar{k}\}] X_p \land [m] ff\]
HML with one recursively defined variable
  - Syntax and semantics
  - How to compute lfp or gfp and when to use them
  - Some temporal properties

Mutually recursive equational systems
  - Syntax and semantics
  - How to compute lfp or gfp

Characteristic properties