# Models and Languages for Computational Systems Biology

## Lecture 8: Mix, match and use temporal logics

Ian Stark

School of Informatics
The University of Edinburgh

Thursday 4 February 2010
Semester 2 Week 4

# Hennessy-Milner Logic

## HML

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False}$$
$$\mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Matthew Hennessy and Robin Milner
On observing nondeterminism and concurrency.
In *Automata, Languages and Programming: Proceedings of the Seventh Colloquium ICALP '80*, Lecture Notes in Computer Science 85, pages 299–309. Springer-Verlag, 1980.

# Hennessy-Milner Logic

## HML

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False}$$
$$\mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Hennessy-Milner logic is a branching-time logic for labelled transition systems.

Like CTL, it can talk about all the possible paths out of a state, and also includes information about the labels on transitions.

Unlike LTL and CTL, it does not make assertions about extended runs of programs, but only single steps at a time.

# Hennessy-Milner Logic

## HML

$$p ::= Atom \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid True \mid False$$
$$\mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Semantics:

| | |
|---|---|
| $s \models \langle a \rangle p$ | There is a transition $s \xrightarrow{a} s'$ for some $s' \models p$. |
| $s \models [a]p$ | For any $s'$, if $s \xrightarrow{a} s'$ then $s' \models p$. |

Note: The $\langle a \rangle$ and $[a]$ echo classic temporal modalities for possibility $\Diamond$ and necessity $\Box$; which are also occasionally used for **A** and G in $CTL/CTL*$; as are $\forall$ and $\exists$.

# Hennessy-Milner Logic

## HML

$$p ::= Atom \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid True \mid False$$
$$\mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Some properties:

$$\neg(\langle a \rangle p) = [a](\neg p)$$
$$[a]p \wedge [a]q = [a](p \wedge q)$$
$$\langle a \rangle p \vee \langle a \rangle q = \langle a \rangle (p \vee q)$$
$$\langle a \rangle p \wedge [a]q \Rightarrow \langle a \rangle (p \wedge q)$$

# Hennessy-Milner Logic

## HML

$$p ::= Atom \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid True \mid False$$
$$\mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Some derived forms:

$$[-]p \stackrel{def}{=} \forall a \in \mathsf{Label} \,.\, [a]p \qquad \langle - \rangle p \stackrel{def}{=} \exists a \in \mathsf{Label} \,.\, \langle a \rangle p$$
$$[-b]p \stackrel{def}{=} \forall a \in (\mathsf{Label} \setminus b) \,.\, [a]p \qquad \langle -b \rangle p \stackrel{def}{=} \exists a \in (\mathsf{Label} \setminus b) \,.\, \langle a \rangle p$$
$$[K]p \stackrel{def}{=} \forall a \in K \subseteq \mathsf{Label} \,.\, [a]p \qquad \langle K \rangle p \stackrel{def}{=} \exists a \in K \subseteq \mathsf{Label} \,.\, \langle a \rangle p$$
$$[-K]p \stackrel{def}{=} \forall a \in (\mathsf{Label} \setminus K) \,.\, [a]p \qquad \langle -K \rangle p \stackrel{def}{=} \exists a \in (\mathsf{Label} \setminus K) \,.\, \langle a \rangle p$$

If the label set is finite then these do not extend HML expressiveness.

# Hennessy-Milner Logic

## HML

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False} \mid \langle a \rangle p \mid [a]p$$

Assertions about states in an LTS and possible transitions between them.

Examples:

| | |
|---|---|
| $s \models [-]\mathrm{False}$ | Deadlock |
| $s \models \langle - \rangle \mathrm{True}$ | $s$ can make some progress |
| $s \models \langle - \rangle \mathrm{True} \wedge [-a]\mathrm{False}$ | $a$ must happen next |
| $s \models \langle a \rangle \mathrm{True} \wedge [a](\langle b \rangle \mathrm{True} \wedge [b][-]\mathrm{False})$ | Transition $a$ then $b$ is possible, but will lead to certain deadlock |

# Modal μ-calculus

## Modal μ

$$p ::= \mathit{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathit{True} \mid \mathit{False}$$
$$\mid \langle a \rangle p \mid [a]p \mid Z \mid \mu X.p \mid \nu Y.p$$

Dexter Kozen
Results on the propositional μ-calculus.
Theoretical Computer Science 27(3):333–354.

# Modal $\mu$-calculus

## Modal $\mu$

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False}$$
$$\mid \langle a \rangle p \mid [a]p \mid Z \mid \mu X.p \mid \nu Y.p$$

Incorporates propositional variables X, Y, Z,. . . , and two fixpoint operators with the following properties:

| | |
|---|---|
| $\mu X.p(X)$ | is the least $M$ such that $M \subseteq p(M)$ |
| $\nu Y.p(y)$ | is the greatest N such that $N \supseteq p(N)$ |

There is an additional constraint that X and Y must be nested within an even number of negations.

Fixpoint logics are notorious for being incomprehensible. (Bradfield & Stirling, 2001)

# Modal μ-calculus

## Modal μ

$$p ::= Atom \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid True \mid False$$
$$\mid \langle a \rangle p \mid [a]p \mid Z \mid \mu X.p \mid \nu Y.p$$

Some equivalences and examples:

$$\neg(\mu X.p(X)) = \nu Y.\neg(p(\neg Y))$$

$$\mathbf{EF}p = \mu X.(p \vee \langle - \rangle X)$$
$$\mathbf{AG}q = \nu Y.(q \wedge [-]Y)$$

$$\mathbf{E}(p \ \mathbf{U} \ q) = \mu X.(q \vee (p \wedge \langle - \rangle X))$$

Modal μ-calculus is sufficiently expressive to encode CTL*, and more besides. (See Bradfield & Stirling 2001 for examples)

# Monadic Second-Order Logic

## MSOL

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False}$$
$$\mid \forall x.p \mid \exists x.p \mid (x \xrightarrow{a} y) \mid Z \mid \forall X.p \mid \exists Y.p$$

Monadic second-order logic is a very general and expressive logic. It uses quantification over states ($\forall x$, $\exists y$) as well as over sets of states ($\forall X$, $\exists Y$), which is sufficient to encode everything we have seen so far.

However, in a precise sense, the modal $\mu$-calculus already captures all distinctions in observable behaviour between systems (although particular behaviours might still be simpler to write in MSOL).

In practice, both modal $\mu$ and MSOL are regarded as hard to read and write, and computationally challenging to verify; although they may form a powerful backend to some more friendly syntax for describing behaviours.

# Mix and Match: CTL−

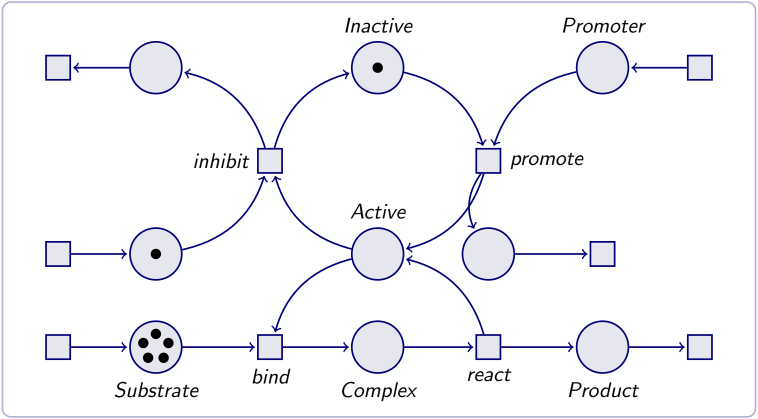
## CTL−

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \mid \mathrm{True} \mid \mathrm{False} \mid \langle a \rangle p \mid [a]p \mid \mathbf{AG}p \mid \mathbf{EF}p \mid \dots$$

One way to make temporal logic more conveniently expressive is to incorporate a range of modalities. Some care may be required to maintain feasibility of model-checking — although comprehensibility is perhaps a deeper problem.

As well as mixing various modalities, the underlying (non-modal) part of the logic may be richer: with assertions over real numbers expressing concentrations of reactants, or quantifiers over numeric values, possibly interlaced with the temporal modalities.

# Example




$$\mathbf{AG}[\mathit{promote}](\mathit{Substrate} \Rightarrow \langle \mathit{complex}\rangle\langle \mathit{react}\rangle \mathit{Product})$$

# Application: LTL on time series

Excerpts from:

F. Fages and A. Rizk.
On the analysis of numerical data time series in temporal logic.
In *Computational Methods in Systems Biology: Proceedings of the International Conference CMSB 2007*, LNBI 4695, pp. 48–63. Springer-Verlag, 2007.

*Reachability* : F([A]>=p), what threshold $p$ species A attain in the trace ?

*Checkpoints* : not (([A]<p1)U([B]>p2)), for which thresholds $p1$ and $p2$ is it false that $[A]$ is lower than $p1$ until $[B]$ is above $p2$, i.e., for which $p1$ and $p2$ [A] >= p1 is a checkpoint of [B]>p2?

*Stability* : G([A]=<p1 & [A]>=p2), what is the range of values taken by $[A]$ ? This range can be looked for in some context given by a condition like in G(Time>10 -> ([A]<p1 & [A]>p2)).

*Oscillation* : F((d([A])/dt>0 & [A]>v1) & (F((d([A])/dt<0 & [A]<v2)))), what amplitude $(v1 - v2)$ is attained in at least one oscillation ? An oscillation is defined as the change of sign of the derivative. This formula can be extended for more oscillations and is abbreviated by oscil(M,K,p). It states that $M$ must have amplitude $P$ in at least $K$ oscillations. By applying the algorithm for each value of $K$, beginning with 1, we can find the number of oscillations in the trace and minimal amplitude $P$ attained by $K$ oscillations for any $K$.

*Influence* : G(d[A]/dt>p1 -> d2[B]/dt2>=0), above which threshold does the derivative of $A$ have an influence on B ? The influence is positive if a high value of d[A]/dt entails a positive second derivative of [B]. It is worth noticing that, as multiple species might influence $B$, this formula only indicates a correlation between the value of the derivative of $A$ and the second derivative of $B$ and gives no proof of direct influence.

For stability, let us find the range of values taken by `[Cdc2]` in the last third part of the trace:

```
biocham: trace_analyze(G(Time>66 -> ([Cdc2]=<v1 & [Cdc2]>=v2))).
[[v1>=0.479, v2=<0.338]]
```

The domain is defined by the conjunction of the two constraints `v1 >= 0.479` and `v2 =< 0.338`. These values are the maximum and minimum values attained by `[Cdc2]` in the last third part of the trace. The results for the other species are given in Table 1.

An oscillation query may compute several interval domains:

```
biocham: trace_analyze(oscil(Cdc2,1)).
[[v2>=0.338, v1=<0.479], [v2>=0.341, v1=<0.479]]
```

The result is the union of two boxes. In such domains, the most relevant point is not obvious. Here we look for the maximum amplitude $v1 - v2$. The maximum is obtained in the domain with $v1 - v2 = 0.479 - 0.338 = 0.141$. This result states that at least one oscillation of `Cdc2` has an amplitude greater or equal to 0.141. The number of oscillations is then incremented until obtaining an empty validity domain. It is obtained for `Cdc2` with the query `oscil(Cdc2,3)`, stating that there are only two oscillations of `Cdc2` in the trace.

# Application: CTL / modal $\mu$ on gene regulation networks


P. T. Monteiro, D. Ropers, R. Mateescu, A. T. Freitas, and H. de Jong.
Temporal logic patterns for querying dynamic models of cellular interaction networks.
*Bioinformatics*, 24(16):227–233, 2008.

H. de Jong.
Qualitative modeling and simulation of bacterial regulatory networks.
*Computational Methods in Systems Biology: Proceedings of the 6th International Conference, CMSB 2008*, Lecture Notes in Bioinformatics 5307, Springer-Verlag, 2008 (with slides)

# References (1/2)

These are the papers introducing CTL*, HML, and the modal μ-calculus, respectively.


E. Allen Emerson and Joseph Y. Halpern
"Sometimes" and "not never" revisited: on branching versus linear time temporal logic.
Journal of the ACM 33(1):151–178.

Matthew Hennessy and Robin Milner
On observing nondeterminism and concurrency.
In *Automata, Languages and Programming: Proceedings of the Seventh Colloquium ICALP '80*, Lecture Notes in Computer Science 85, pages 299–309. Springer-Verlag, 1980.

Dexter Kozen
Results on the propositional μ-calculus.
Theoretical Computer Science 27(3):333–354.

# References (2/2)

Bradfield and Stirling give a good overview of temporal logics and model-checking; Vardi argues for the comprehensibility of LTL over CTL; and Janin and Walukiewicz show that the modal μ-calculus is all you need for observable behaviour.


Julian Bradfield and Colin Stirling
Modal logics and mu-calculi: an introduction.
In *Handbook of Process Algebra*, pages 293–330. Elsevier, 2001.

Moshe Vardi
Branching vs. Linear Time: Final Showdown
In *Proc. TACAS 2001*, LNCS 2031, pp. 1–22. Springer-Verlag, 2001.

David Janin and Igor Walukiewicz
On the expressive completeness of the propositional mu-calculus with respect to monadic second-order logic.
In *Proc. Concur '96*, LNCS 1119, pp. 263–277. Springer-Verlag, 1996.