Models and Languages for Computational Systems Biology

Lecture 7: From LTL and CTL to CTL*

Ian Stark

School of Informatics
The University of Edinburgh

Monday 1 February 2010
Semester 2 Week 4

LTL vs. CTL

Linear Temporal Logic

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p$$
$$\mathbf{X}p \mid \mathbf{F}p \mid \mathbf{G}p \mid p\,\mathbf{U}\,q \mid p\,\mathbf{R}\,q$$

Assertions over individual runs of a transition system.

Computation Tree Logic

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p$$
$$\mathbf{AX}p \mid \mathbf{AF}p \mid \mathbf{AG}p \mid \mathbf{A}(p\,\mathbf{U}\,q) \mid \mathbf{A}(p\,\mathbf{R}\,q)$$
$$\mathbf{EX}p \mid \mathbf{EF}p \mid \mathbf{EG}p \mid \mathbf{E}(p\,\mathbf{U}\,q) \mid \mathbf{E}(p\,\mathbf{R}\,q)$$

Assertions over the tree of all possible runs of a transition system.

LTL vs. CTL

Linear Temporal Logic

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p$$
$$\mathbf{X}p \mid \mathbf{F}p \mid \mathbf{G}p \mid p\,\mathbf{U}\,q \mid p\,\mathbf{R}\,q$$

Assertions over individual runs of a transition system.

- LTL is *linear time*, looking only at a single observed run.
- Often applied to all possible runs from a given initial state.
- Can express safety ($\mathbf{G}(\neg \mathit{Bad})$) and liveness ($\mathbf{GF}(\mathit{Enabled})$).
- Model-checking tools can operate on state machines, simulated runs, or real time-series data.

e.g. [Fages & Rizk 2007]

Computation Tree Logic

Computation Tree Logic

$$\begin{aligned} p ::= {} & \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p \\ & \mathbf{AX}p \mid \mathbf{AF}p \mid \mathbf{AG}p \mid \mathbf{A}(p\ \mathbf{U}\ q) \mid \mathbf{A}(p\ \mathbf{R}\ q) \\ & \mathbf{EX}p \mid \mathbf{EF}p \mid \mathbf{EG}p \mid \mathbf{E}(p\ \mathbf{U}\ q) \mid \mathbf{E}(p\ \mathbf{R}\ q) \end{aligned}$$

Atom	Some basic predicate over system states
$\mathbf{AX}p$	p holds after any possible single step
$\mathbf{AF}p$	p holds at some later state on any run
$\mathbf{AG}p$	p holds at all states on all runs from here
$\mathbf{A}(p\ \mathbf{U}\ q)$	for all runs, p holds until finally q does
$\mathbf{A}(p\ \mathbf{R}\ q)$	for all runs, q holds indefinitely or until p also does

Computation Tree Logic

Computation Tree Logic

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p$$
$$\mathbf{AX}p \mid \mathbf{AF}p \mid \mathbf{AG}p \mid \mathbf{A}(p\ \mathbf{U}\ q) \mid \mathbf{A}(p\ \mathbf{R}\ q)$$
$$\mathbf{EX}p \mid \mathbf{EF}p \mid \mathbf{EG}p \mid \mathbf{E}(p\ \mathbf{U}\ q) \mid \mathbf{E}(p\ \mathbf{R}\ q)$$

Atom	Some basic predicate over system states
$\mathbf{EX}p$	p holds after some possible single step
$\mathbf{EF}p$	p holds at some later state on some run
$\mathbf{EG}p$	p holds at all states on some run from here
$\mathbf{E}(p\ \mathbf{U}\ q)$	for some run, p holds until finally q does
$\mathbf{E}(p\ \mathbf{R}\ q)$	for some run, q holds indefinitely or until p also does

Computation Tree Logic

Computation Tree Logic

$$p ::= \mathrm{Atom} \mid p \wedge q \mid p \vee q \mid p \Rightarrow q \mid \neg p$$
$$\mathbf{AX}p \mid \mathbf{AF}p \mid \mathbf{AG}p \mid \mathbf{A}(p\ \mathbf{U}\ q) \mid \mathbf{A}(p\ \mathbf{R}\ q)$$
$$\mathbf{EX}p \mid \mathbf{EF}p \mid \mathbf{EG}p \mid \mathbf{E}(p\ \mathbf{U}\ q) \mid \mathbf{E}(p\ \mathbf{R}\ q)$$

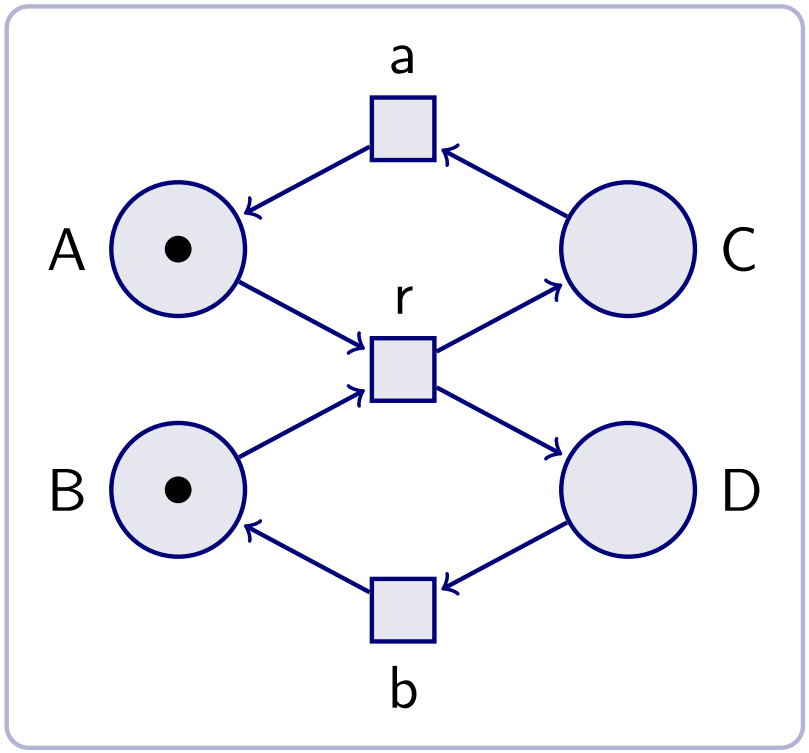
$$\neg(\mathbf{AX}p) = \mathbf{EX}(\neg p)$$
$$\neg(\mathbf{AF}p) = \mathbf{EG}(\neg p)$$
$$\neg(\mathbf{AG}p) = \mathbf{EF}(\neg p)$$

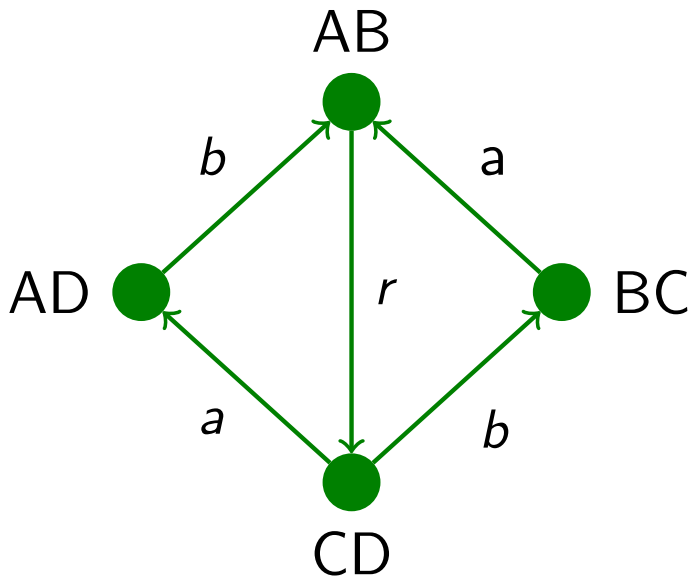
$$\neg(\mathbf{A}(p\ \mathbf{U}\ q)) = \mathbf{E}((\neg p)\ \mathbf{R}\ (\neg q))$$
$$\neg(\mathbf{A}(p\ \mathbf{R}\ q)) = \mathbf{E}((\neg p)\ \mathbf{U}\ (\neg q))$$

$$\mathbf{AF}p = \mathbf{A}(\mathrm{True}\ \mathbf{U}\ p)$$
$$\mathbf{AG}p = \mathbf{A}(p\ \mathbf{R}\ \mathrm{False})$$

So **AX**, **AU**, **AR** provide a complete set of connectives.

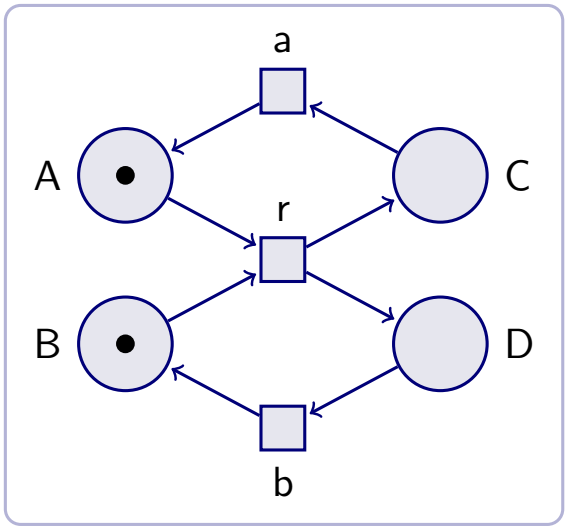
CTL valuation

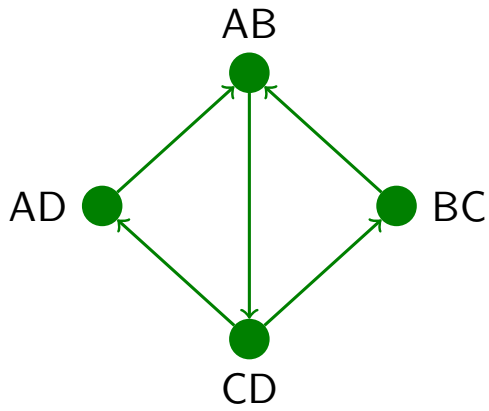




Identify at which states the CTL formula $\mathbf{AF}(\mathbf{EX}(\mathrm{AD}) \wedge \mathbf{EX}(\mathrm{BC}))$ holds.

CTL valuation





	AB	CD	AD	BC
AD	✗	✗	✓	✗
BC	✗	✗	✗	✓
EX(AD)	✗	✓	✗	✗
EX(BC)	✗	✓	✗	✗
EX(AD) ∧ **EX**(BC)	✗	✓	✗	✗
AF(**EX**(AD) ∧ **EX**(BC))	✓	✓	✓	✓

CTL*

Formulae of CTL*

State formula $\quad p ::= \mathrm{Atom} \mid p \wedge p' \mid p \vee p' \mid p \Rightarrow p' \mid \neg p$
$\qquad\qquad\qquad\qquad \mathbf{A}q \mid \mathbf{E}q$

Path formula $\quad q ::= p \mid q \wedge q' \mid q \vee q' \mid q \Rightarrow q' \mid \neg q'$
$\qquad\qquad\qquad\qquad \mathbf{X}q \mid \mathbf{F}q \mid \mathbf{G}q \mid q\,\mathbf{U}\,q' \mid q\,\mathbf{R}\,q'$

The branching-time logic *CTL** combines both state formulae of CTL and path formulae of LTL, with the possibility to switch between them at any point in an expression.

CTL*

Formulae of CTL*

State formula $\quad p ::= \mathrm{Atom} \mid p \wedge p' \mid p \vee p' \mid p \Rightarrow p' \mid \neg p$
$\qquad\qquad\qquad\qquad \mathbf{A}q \mid \mathbf{E}q$

Path formula $\quad q ::= p \mid q \wedge q' \mid q \vee q' \mid q \Rightarrow q' \mid \neg q'$
$\qquad\qquad\qquad\qquad \mathbf{X}q \mid \mathbf{F}q \mid \mathbf{G}q \mid q\,\mathbf{U}\,q' \mid q\,\mathbf{R}\,q'$

$s \models p \wedge p'$ iff $s \models p$ and $s \models p'$

$s \models p \vee p'$ iff $s \models p$ or $s \models p'$

etc.

$s \models \mathbf{A}q$ iff $r \models q$ for all runs $r = s\,s'\,s''\ldots$

$s \models \mathbf{E}q$ iff $r \models q$ for some run $r = s\,s'\,s''\ldots$

CTL*

Formulae of CTL*

State formula $\quad p ::= \mathrm{Atom} \mid p \wedge p' \mid p \vee p' \mid p \Rightarrow p' \mid \neg p$
$\qquad\qquad\qquad\qquad \mathbf{A}q \mid \mathbf{E}q$

Path formula $\quad q ::= p \mid q \wedge q' \mid q \vee q' \mid q \Rightarrow q' \mid \neg q'$
$\qquad\qquad\qquad\qquad \mathbf{X}q \mid \mathbf{F}q \mid \mathbf{G}q \mid q\,\mathbf{U}\,q' \mid q\,\mathbf{R}\,q'$

$r \models p$ iff $s \models p$ where $r = s\,s'\,s''\ldots$

$r \models \mathbf{X}q$ iff $r' \models q$ where $r = sr'$

$r \models \mathbf{F}q$ iff $r' \models q$ for some r' with $r = s\,s'\,s''\ldots r'$

$r \models \mathbf{G}q$ iff $r' \models q$ for all r' with $r = s\,s'\,s''\ldots r'$

$r \models q\,\mathbf{U}\,q$ *etc.*

CTL*

Formulae of CTL*

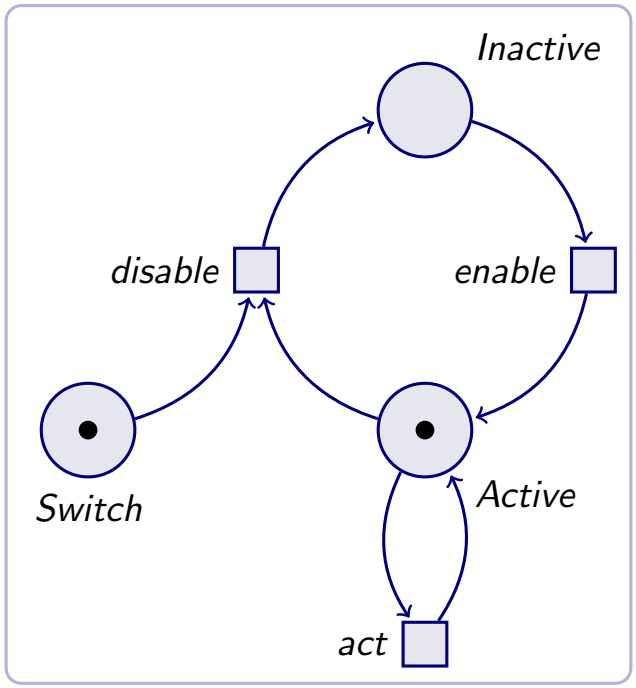
State formula $\quad p ::= \mathrm{Atom} \mid p \wedge p' \mid p \vee p' \mid p \Rightarrow p' \mid \neg p$
$\qquad\qquad\qquad\qquad \mathbf{A}q \mid \mathbf{E}q$

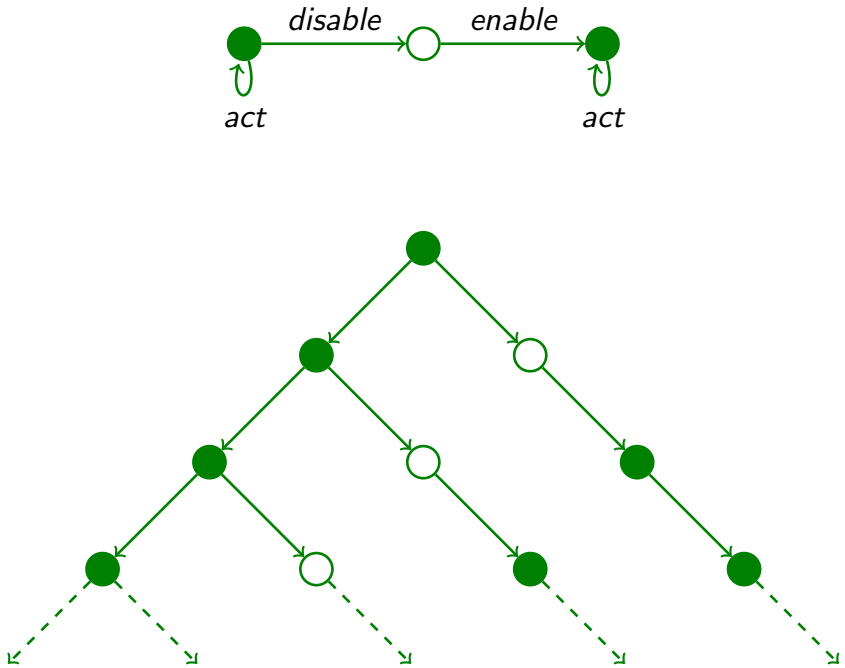
Path formula $\quad q ::= p \mid q \wedge q' \mid q \vee q' \mid q \Rightarrow q' \mid \neg q'$
$\qquad\qquad\qquad\qquad \mathbf{X}q \mid \mathbf{F}q \mid \mathbf{G}q \mid q\,\mathbf{U}\,q' \mid q\,\mathbf{R}\,q'$

Every CTL formula is a CTL* formula, and (prefixed with **A**) so is every LTL formula. Both are proper subsets, and neither CTL nor LTL contain the other.

For example, the LTL formula $\mathbf{FG}(p)$, represented in CTL* as $\mathbf{AFG}(p)$, is not expressible in CTL. In particular, it is different from $\mathbf{AFAG}(p)$.

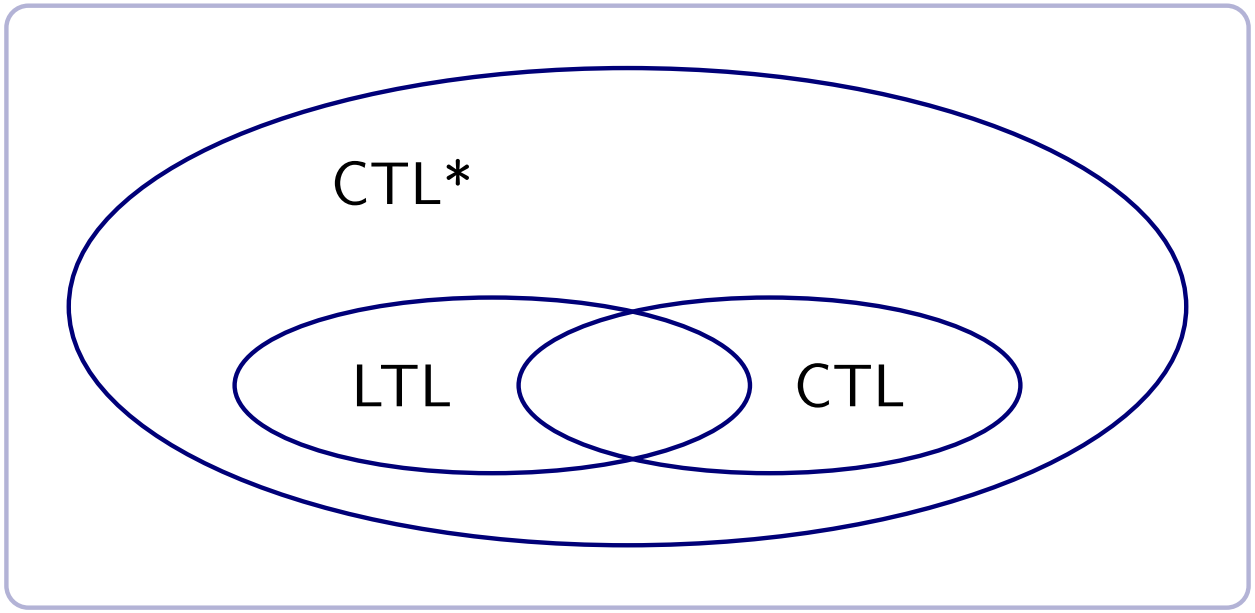
Why FG is not AF AG



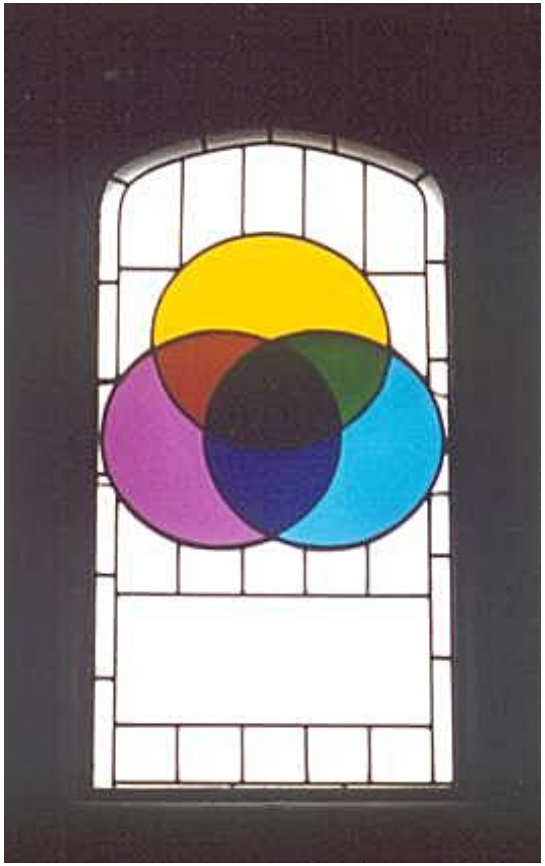


Every run of this Petri net satisfies the LTL formula **FG**(*Active*); but its initial marking does not satisfy the CTL formula **AFAG**(*Active*).

LTL and CTL in CTL*



On Venn diagrams



The Venn window in the dining hall of
Gonville & Caius College, Cambridge

Reading

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.
Proc. CMSB 2008, Lecture Notes in Bioinformatics 5307, Springer-Verlag, 2008 (with slides)