6894 : Lightweight Formal Methods.

Lecture 12 : Temporal Logic, Part II.

Daniel Jackson March 20, 2005.

Last time:

~~Kripke~~

Kripke structure:

state machine

states labelled w/ sets of propositions

no dead ends — transition from every state

computation tree

unfolded state machine.

CTL*

generalization of LTL and CTL

state formulas: ϕ

P, $A\alpha$, $E\alpha$

path formulas

$F\alpha$, $G\alpha$, $\alpha \, U \, \alpha$, $X\alpha$

every state formula is also a path formula

Weak until.

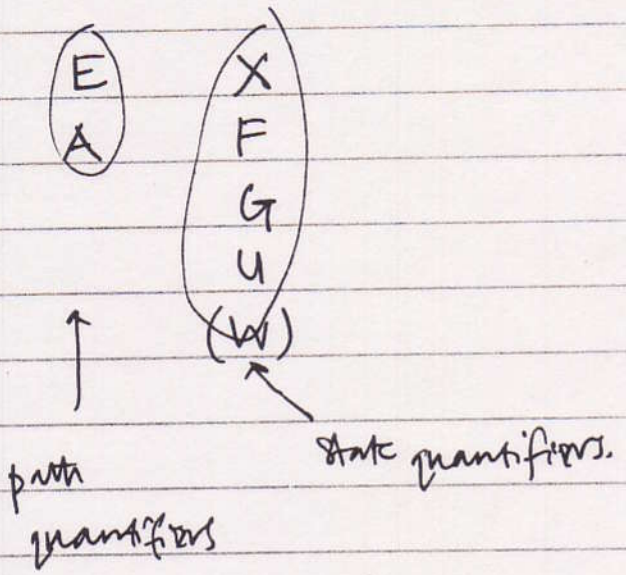
$f \, U \, g$ says g eventually holds, and until then, f holds.

$f \, W \, g$ says either f holds for ever, or holds until g ho

CTL.

subset of CTL*

idea: temporal operators occur only in pairs.



grammar is now:

formula = f

::= p | ¬f | f ∨ f | f ∧ f

EX f | EF f | EG f | ...

meanings

EX p — p may hold in next step

AX p — must

EF p — p may occur eventually

AG p — p holds always.

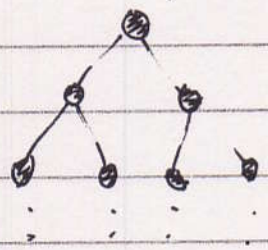
EX, EG and EU are enough!

$$AX\,f = \neg EX(\neg f)$$
$$AG\,f = \neg EF(\neg f)$$
$$AF\,f = \neg EG(\neg f)$$
$$EF\,f = E[true\ U\ f]$$
$$A(f\ U\ g) = \neg E[\neg g\ U\ (\neg f \wedge \neg g)] \wedge \neg EG\,\neg g.$$

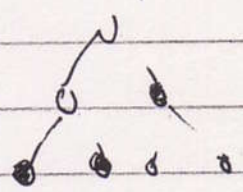
Note use of [ ] to find U operator.

4 most common operators.

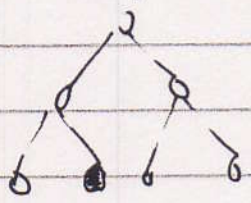
AG p



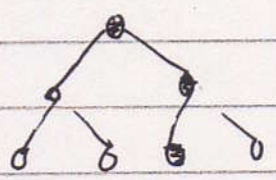
AF p



EF p



EG p



Sample CTL formulas

$AG \neg (Green_{NS} \wedge Green_{WE})$

$AG\,(Req \Rightarrow AF\,Ack)$

$AG(EF\ Restart)$

maybe omit this, since expressive ← FL

Some more complicated patterns ← Skip these.

KSH
(from "Spec Patterns", SANTOS)

~~Absence~~ Universality

P is ~~false~~ true

globally	$AG(\neg P)$
before R	$A[(\neg P \vee (AG \neg R))\ W\ R]$
after Q	$AG(Q \Rightarrow AG(\neg P))$

Existence

P becomes true

globally	$AF\ P$
before R	$A[\neg R\ W\ (P \wedge \neg R)]$
after Q	$A[\neg Q\ W\ (Q \wedge AF\ P)]$

Precedence.

S precedes P

globally	$A\,[\neg P\ W\ S]$
before R	$A\,[(\neg P \vee AG\neg R,\ W\ (S \vee R)]$
after Q	$A\,[\neg Q\ W\ (Q \wedge A\,[\neg P\ W\ S])]$

Response.

S responds to P

globally	$AG(? \Rightarrow AFS)$
before R	(complicated)
after Q	$A\,[\neg Q\ W\ (Q \wedge AG(P \Rightarrow AFS))]$

Exercises for class on CTL.

(printer example).

LTL.

LTL subset of CTL*.
Only path quantifier is A,
only allowed once outermost.

formula ::= A path formula
path formula α ::=
$p \mid F\alpha \mid G\alpha \mid \alpha \, U \, \alpha \mid X\alpha$

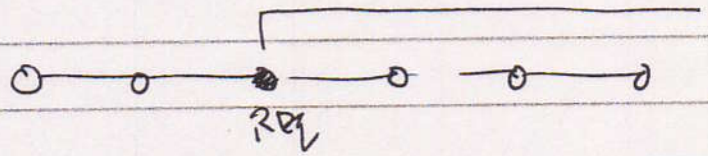
In LTL, different symbols used.
- outermost A dropped.
- $\Box f$ Gf
  $\Diamond f$ Ff
  $\circ f$ Xf

~~Hyperproperties~~

Examples.

$\Box \neg (Green_{NS} \wedge Green_{EW})$

$\Box (Req \Rightarrow \Diamond Resp)$



Spec Patterns for LTL (not discussed in class)

~~Absence~~ - Universality

P is ~~false~~ true

globally	$\Box \neg P$
before R	$\Diamond R \Rightarrow \neg P \; U \; R$
after Q	$\Box (Q \Rightarrow \Box \neg P)$

Existence:

P becomes true

globally	$\Diamond P$
before R	$\neg R \; W \; (P \wedge \neg R)$
after Q	$\Box (\neg Q \vee \Diamond (Q \wedge \Diamond P))$
	$\equiv \Box (Q \Rightarrow \Diamond P)$?

Precedence

No, they mean $(\Box \neg Q) \vee \Diamond (Q \wedge \Diamond P)$

S precedes P

globally	$\neg P \; W \; S$
before R	$\Diamond R \Rightarrow (\neg P \; U \; (S \vee R))$
after Q	$(\Box \neg Q) \vee \Diamond (Q \wedge (\neg P \; W \; S))$

Response

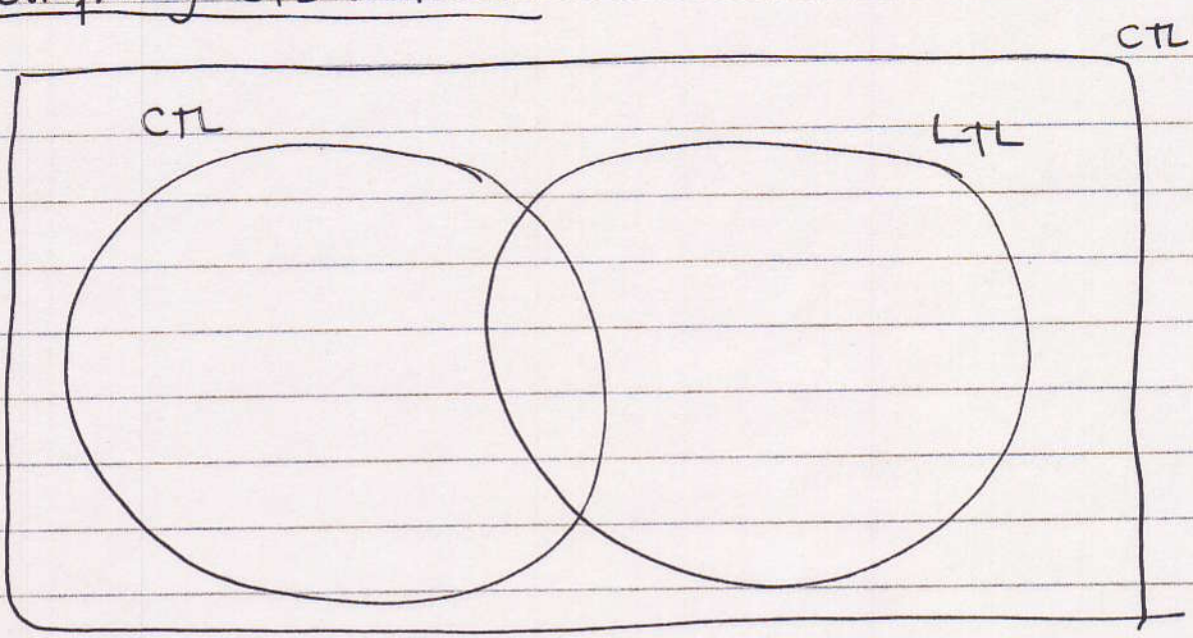
S responds to P

globally	$\Box (P \Rightarrow \Diamond S)$
before R	$\Diamond R \Rightarrow (P \Rightarrow (\neg R \vee (S \wedge \neg R))) \; U \; R$
after Q	$\Box Q \Rightarrow (\Box (P \Rightarrow \Diamond S))$

Exercises for class in LTL

(Printer)

Comparing CTL and LTL



Comparing them is subtle.

is $AFAGp \equiv AFGp$? [No]

actually $AFGp$ expressible ∈ CTL.

In LTL but not CTL:

fairness constraints

$A(GFp \Rightarrow Fq)$

so have to 'hardwire' fairness into CTL-based model checkers.

In CTL but not LTL

$AGEFp$ — p is always possible.

Can express deadlock avoidance like this.

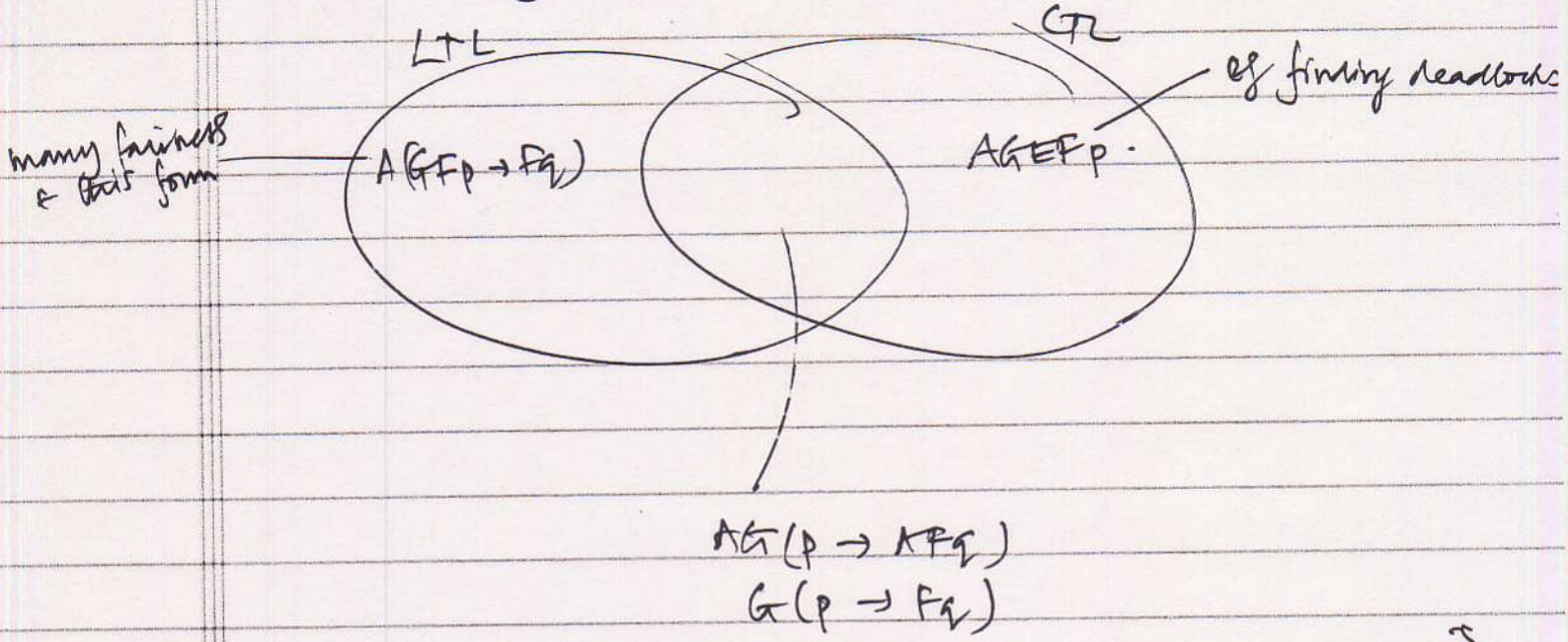
Proof that $AGEFp \notin$ LTL: see attached sheet.

Generally, LTL ~~=~~ easier than CTL

but (sometimes) less tractable

abc def.

from Huth/Ryan

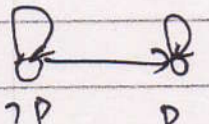


$FGp \neq AFAGp$

$E[GF_p] \in LTL \vee CTL$..

didn't show this proof, but used the machine to show $AGEFp \neq AGFp$

proof that $AGEFp \notin LTL$:
Let ϕ sat $A\phi \equiv AGEFp$.



$M \models AGEFp$

So $M \not\models A\phi$

consider  M'

paths $M' \subseteq$ paths M.
so $M' \models A\phi$
but $M' \not\models AGEFp$.

"Sometime" is sometimes "Not never"
(Lamport, 1980).

In LTL:

Not never P is $\neg \Box \neg P \equiv \Diamond P$

which is 'sometime P'

But in CTL:

Not never P is $\neg AG \neg P \equiv EF P$

which is <u>possibly P</u>, not sometime P.

Safety + Liveness

Informally:

Safety: something bad does ¬ happen.

Liveness: something good eventually happens.

More formally:

F is a safety formula iff
any sequence π violating F has a finite prefix π'
all of whose ω extensions violate F

F is a liveness property iff
any finite seq π can be extended to an ω seq satisfying F.

Theorem:
almost disjoint!
only true $\in$ safety $\cap$ liveness.

Examples.

safety	global inv	$\Box p$
	local i'nv	$\Box (pc = \ell \Rightarrow p)$
	partial correctness	$\Box (pc = RET \rightarrow p)$
	deadlock freedom	$\Box \neg$ terminal (uses special pred

Livelock: not deadlocked, but one process stuck at ℓ:
$\Box \Diamond \neg (pc = \ell)$

↓
liveness, see below.

Liveness examples.

termination $\Diamond(pc = RET)$

livelock absence.

eventual reliability $\Box \Diamond Req \Rightarrow \Diamond Resp$.

Fairness.

Suppose we have a 2 process system (Requester / Responder), and we write a liveness property such as

$$AG(Req \Rightarrow AF\ Resp)$$

and we get a counterex in which the responder never even got scheduled!

In LTL, we can filter the executions:

$$\Box\Diamond\ RespRuns \Rightarrow \Box\ Req \Rightarrow \Diamond\ Resp$$

~~But $\Box\Diamond p \notin CTL$!~~

~~$\Box\Diamond p = AGFp \neq AG\,AF\,p$.~~

In CTL*:

$$A\,(GF\,p \Rightarrow F\,Q)$$

Not same as

$$AG\,(AF\,p \Rightarrow AF\,Q)$$

because $\exists$ unfair executions!

So CTL model checkers must hardwire fairness.
Consider only execs wrt a fairness constraint
> non-temporal property p is true infinitely often

Printer.

$Queued_j$

$Printed_j$

Ack_j

a job that is queued is eventually printed:

$AG(Q_j \Rightarrow AF P_j)$ $\quad\quad$ $\Box(Q_j \Rightarrow \Diamond P_j)$

an ack is never sent unless a job was printed:

$A[\neg Ack_j \; W \; (P_j \wedge \neg Ack_j)]$

$\neg Ack_j \; W \; (P_j \wedge \neg Ack_j)$

the same job is ack at most once

$AG \; Ack_j \Rightarrow \overset{AX}{AG} \neg Ack_j$. $\quad\quad$ $\Box A_j \Rightarrow \bigcirc \Box \neg A_j$

if a job is queued infinitely often, it will eventually be printed

$(\Box \Diamond Q_j) \Rightarrow \Diamond P_j$