Branching-time logic CTL

Sintax

$\varphi ::= \ldots\ldots$

$$AX\varphi \mid EX\varphi \mid AG\varphi \mid AF\varphi$$
$$EG\varphi \mid EF\varphi \mid A[\varphi U \varphi] \mid E[\varphi U \varphi]$$

Convention: connectives of CTL bind in the following order

unary : $(\neg, AG, EG, AF, EF, AX, EX)$

then $(\wedge, \vee)$

finally $(\rightarrow, AU, EU)$

Example: WFF?

$AG(q \rightarrow EGr)$ ✓

$AGq \rightarrow EGr$ ✓

$EF(EGp)$ ✓

$EF(rUq)$ ✗

$A[rUq \vee pUq]$ ✗

$AEFr)$ ✗

?

$A[(r \vee q)U(p \vee q)]$ ✓

$A[r \vee (qUr) \vee q]$ ✗

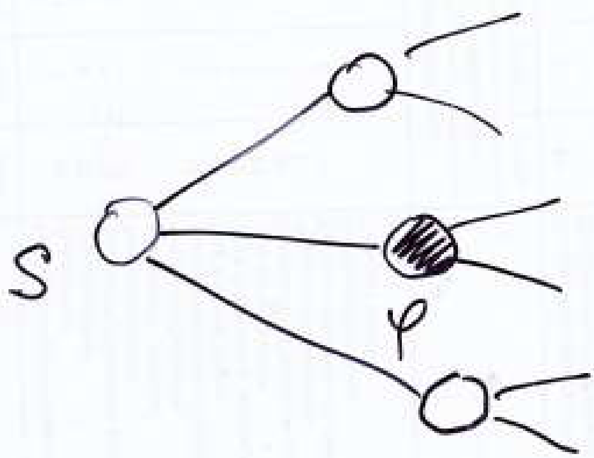
AU? $A[\varphi U \varphi]$

Semantics

$M, s \models EX\varphi$ iff

for some s_1 s.t. $s \to s_1$

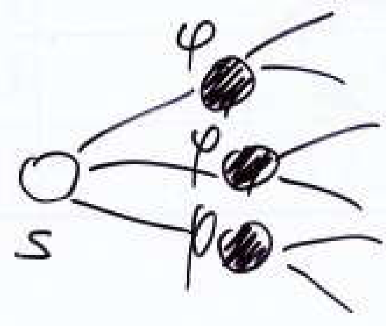
$M, s_1 \models \varphi$



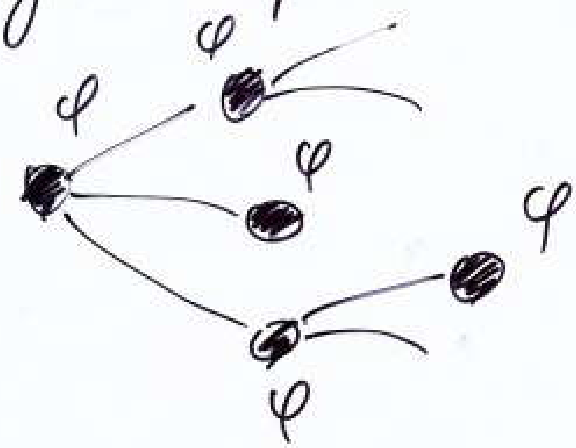
$M, s \models AX\varphi$ iff

for all s_1 s.t. $s \to s_1$

$M, s_1 \models \varphi$

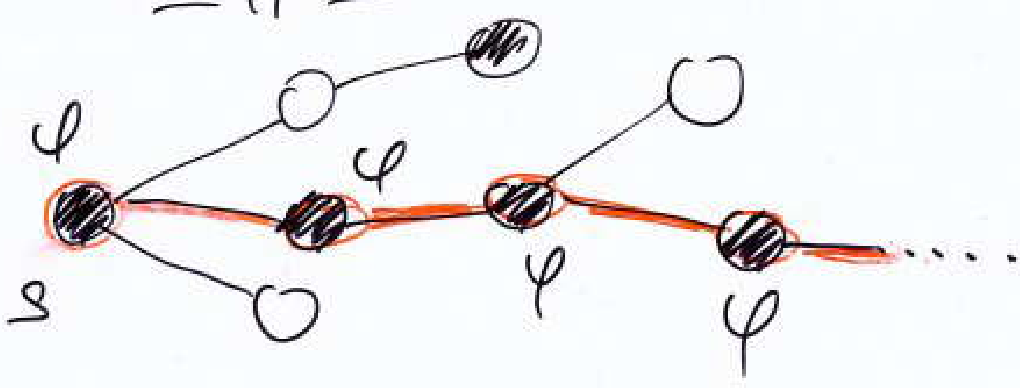


$M, s \models AG\varphi$ iff for all paths $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$ where $s = s_1$ and for all states s_i along the path, $M, s_i \models \varphi$

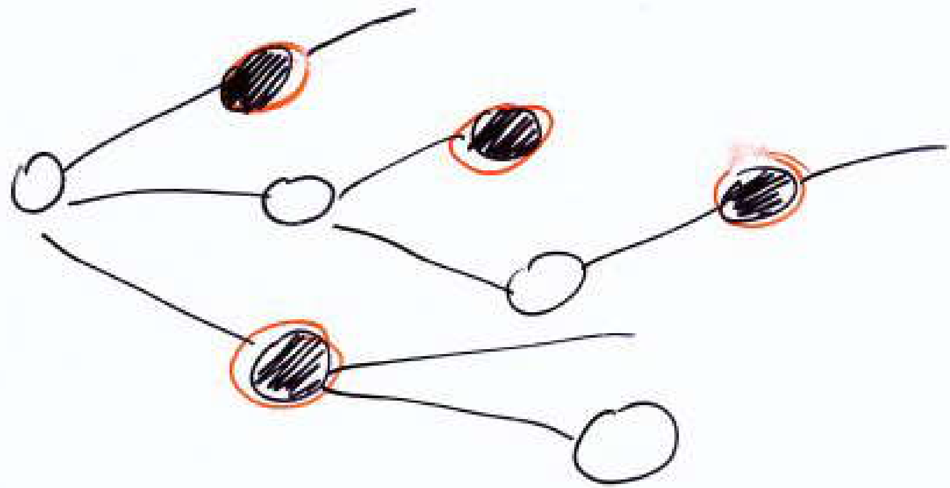


$M, s \models EG\varphi$ iff exists

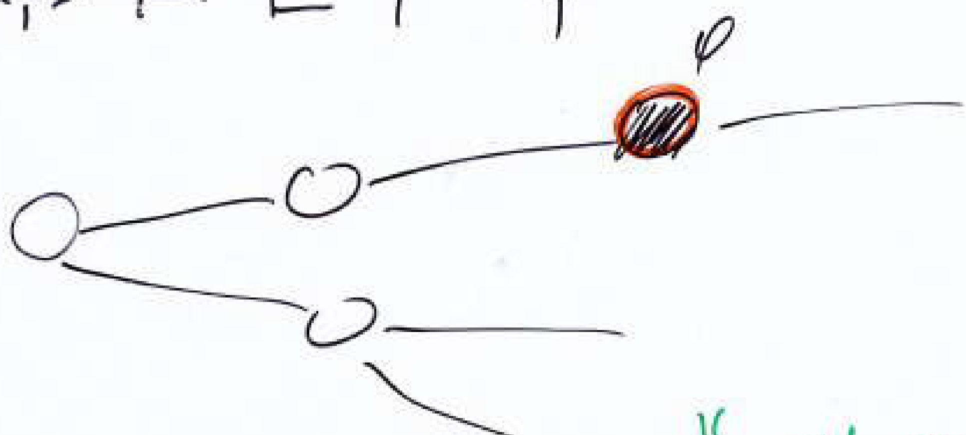
— " —



$M, s \models AF\,\varphi$

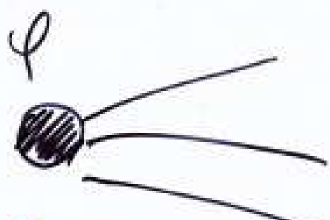


$M, s \models EF\,\varphi$



"path quantifie
A, E
"state quantife
G, F

In particular,



(for both AF, EF)

$M, s \models A[\varphi_1 \, \mathcal{U} \, \varphi_2]$ iff

In *every* path,

In particular

$M, s \models E[\varphi_1 \, \mathcal{U} \, \varphi_2]$ iff

In some path,

CTL examples

$M, s_0 \models AX(AF\,r)$ ✓

$M, s_0 \models AX(AG\,r)$ ✗

$M, s_0 \models EX(AG\,r)$ ✓

$M, s_0 \models q \rightarrow EX\,r$ ✓

$M, s_0 \models (p \wedge q) \wedge AX(A[p\,U\,r])$ ✓

$M, s_0 \models AG\,p \rightarrow p$ ✓

$M, s_1 \models EX(AG\,r)$ ✓

$M, s_0 \models EG\,E[p\,U\,r]$ ✓

LTL is more expressive than CTL

Can describe paths with several conditions along the same path, e.g.

"if π has p then π has q as well"

$$Fp \rightarrow Fq$$

CTL is more expressive than LTL:

Can represent branching time properties, e.g. "for every state satisfying p there is a successor satisfying q"

Can we combine the advantages of both?

Yes! CTL* !

Idea: drop the CTL constraint that each of (X, U, F, G) must be associated with a unique path quantifier

Can write:

$A\,[(p\,U\,r) \vee (q\,U\,r)]$

$\not\equiv A\,[(p \vee q)\,U\,r]$

$E\,[G\,E\,p]$

$\not\equiv EGEFp$

CTL*

Syntax:

- State formulas (evaluated in states)

$$\varphi ::= \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid$$
$$A[\alpha] \mid E[\alpha]$$

where α is a path formula

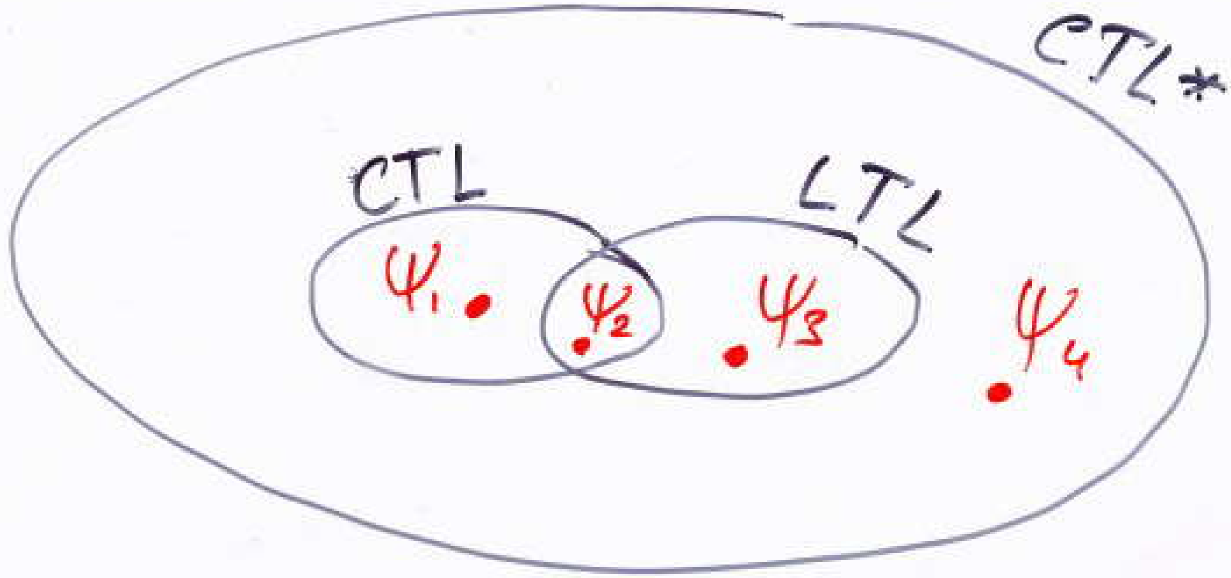
- Path formulas (evaluated along paths)

$$\alpha ::= \varphi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha \, U \, \alpha) \mid$$
$$(G\alpha) \mid (F\alpha) \mid (X\alpha),$$

where φ is a state formula

(mutially recursive def.)

LTL and CTL as subsets of CTL*



In CTL, not in LTL $\quad \psi_1 = AGEF\,p$

In LTL and in CTL $\quad \psi_2 = G(p \rightarrow Fq)$

✓ $\stackrel{?}{=} AG(p \rightarrow AFq)$

In LTL, not in CTL $\quad \psi_3 = A[G(F\,p) \rightarrow Fq]$

inf. often p → an occu[rrence] of q

Emerson (hard)

In CTL*, not in CTL, not in LTL $\quad \psi_4 = E\,[GF\,p]$

$AG(n_1 \to EXt_1)$ is not expressible in LTL

Proof: Sup. it is expr. by φ,

then $AG(n_1 \to EXt_1) \equiv A[\varphi]$ (φ: LTL) (in CTL*)

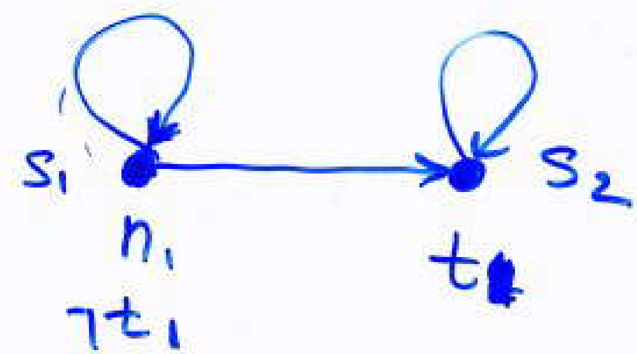
$\forall M \forall s$

$M, s \models AG(n_1 \to EXt_1)$

iff

$M, s \models A[\varphi]$

M_1:



Since $M_1, s_1 \models \boxed{AG\,(n_1 \to EX\,t_1)}$,

we have $M_1, s_1 \models A[\varphi]$

Consider:

M_2:

Since $M_1, s_1 \models A[\varphi]$, we must have $M_2, s_1 \models A[\varphi]$.

But $M_2, s_1 \not\models AG\,(n_1 \to EX\,t_1)$!

contradiction.