Formal Verification

Lecture 5: Computation Tree Logic (CTL)

Jacques Fleuriot[1]
jdf@inf.ac.uk

[1] With thanks to Bob Atkey for some of the diagrams.

Recap

- Previously:
  - *Linear-time* Temporal Logic
- This time:
  - A *branching-time* logic: Computation Tree Logic (CTL)
  - Syntax and Semantics
  - Comparison with LTL, CTL*
  - Model checking CTL

CTL Syntax

Assume a set *Atom* of atom propositions.

$$
\begin{aligned}
\phi \;::=\;& p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \\
& \mid \mathbf{AX}\ \phi \mid \mathbf{EX}\ \phi \mid \mathbf{AF}\ \phi \mid \mathbf{EF}\ \phi \mid \mathbf{AG}\ \phi \mid \mathbf{EG}\ \phi \\
& \mid \mathbf{A}[\phi\ \mathbf{U}\ \phi] \mid \mathbf{E}[\phi\ \mathbf{U}\ \phi]
\end{aligned}
$$

where $p \in Atom$.

Each temporal connective is a pair of a *path quantifier*:

A – for all paths

E – there exists a path

and an LTL-like temporal operator **X**, **F**, **G**, **U**.

Precedence (high-to-low): $(\mathbf{AX}, \mathbf{EX}, \mathbf{AF}, \mathbf{EF}, \mathbf{AG}, \mathbf{EG}, \neg), (\wedge, \vee), \rightarrow$

CTL Semantics 1: Transition Systems and Paths

(*This is the same as for LTL*)

Definition (Transition System)

A *transition system* $\mathcal{M} = \langle S, \rightarrow, L \rangle$ consists of:

$$\begin{array}{cl} S & \text{a finite set of states} \\ \rightarrow \subseteq S \times S & \text{transition relation} \\ L : S \rightarrow \mathcal{P}(Atom) & \text{a labelling function} \end{array}$$

such that $\forall s_1 \in S.\ \exists s_2 \in S.\ s_1 \rightarrow s_2$

Definition (Path)

A *path* π in a transition system $\mathcal{M} = \langle S, \rightarrow, L \rangle$ is an infinite sequence of states $s_0, s_1, \ldots$ such that $\forall i \geq 0.\ s_i \rightarrow s_{i+1}$.

Paths are written as: $\pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$

CTL Semantics 2: Satisfaction Relation

Satisfaction relation $\mathcal{M}, s \models \phi$ read as

state s in model $\mathcal{M}$ satisfies CTL formula ϕ

We often leave $\mathcal{M}$ implicit.
The propositional connectives:

$$
\begin{array}{llllll}
s & \models & \top & & \\
s & \not\models & \bot & & \\
s & \models & p & \text{iff} & p \in L(s) \\
s & \models & \neg\phi & \text{iff} & s \not\models \phi \\
s & \models & \phi \wedge \psi & \text{iff} & s \models \phi \text{ and } s \models \psi \\
s & \models & \phi \vee \psi & \text{iff} & s \models \phi \text{ or } s \models \psi \\
s & \models & \phi \rightarrow \psi & \text{iff} & s \models \phi \text{ implies } s \models \psi
\end{array}
$$

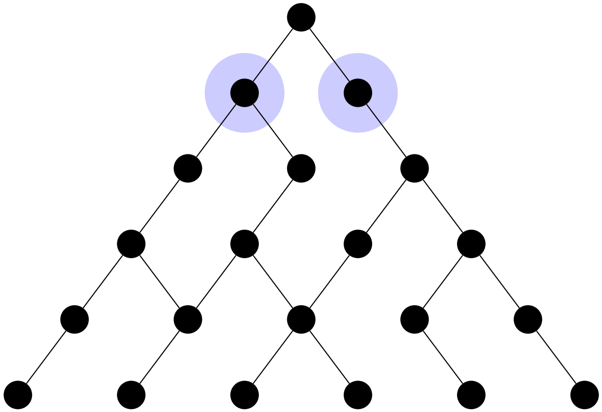
CTL Semantics 2: Satisfaction Relation

The temporal connectives, assuming path $\pi = s_0 \to s_1 \to s_2 \to \ldots$,

$$
\begin{array}{llll}
s & \models & \mathbf{AX}\,\phi & \text{iff} \quad \forall\pi \text{ s.t. } s_0 = s.\ s_1 \models \phi \\
s & \models & \mathbf{EX}\,\phi & \text{iff} \quad \exists\pi \text{ s.t. } s_0 = s.\ s_1 \models \phi \\
s & \models & \mathbf{AG}\,\phi & \text{iff} \quad \forall\pi \text{ s.t. } s_0 = s.\ \forall i.\ s_i \models \phi \\
s & \models & \mathbf{EG}\,\phi & \text{iff} \quad \exists\pi \text{ s.t. } s_0 = s.\ \forall i.\ s_i \models \phi \\
s & \models & \mathbf{AF}\,\phi & \text{iff} \quad \forall\pi \text{ s.t. } s_0 = s.\ \exists i.\ s_i \models \phi \\
s & \models & \mathbf{EF}\,\phi & \text{iff} \quad \exists\pi \text{ s.t. } s_0 = s.\ \exists i.\ s_i \models \phi \\
s & \models & \mathbf{A}[\phi\ \mathbf{U}\ \psi] & \text{iff} \quad \forall\pi \text{ s.t. } s_0 = s. \\
 & & & \qquad\quad \exists i.\ s_i \models \psi \text{ and } \forall j < i.\ s_j \models \phi \\
s & \models & \mathbf{E}[\phi\ \mathbf{U}\ \psi] & \text{iff} \quad \exists\pi \text{ s.t. } s_0 = s. \\
 & & & \qquad\quad \exists i.\ s_i \models \psi \text{ and } \forall j < i.\ s_j \models \phi
\end{array}
$$

Note: The semantics for **AX** and **EX** is given differenttly in H&R.

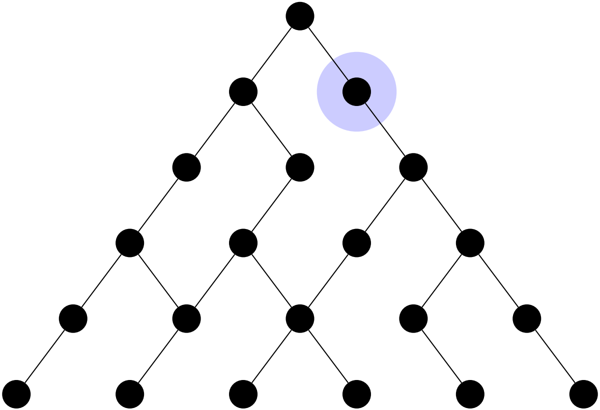
CTL in Pictures



AX ϕ

For *every* next state, ϕ holds.

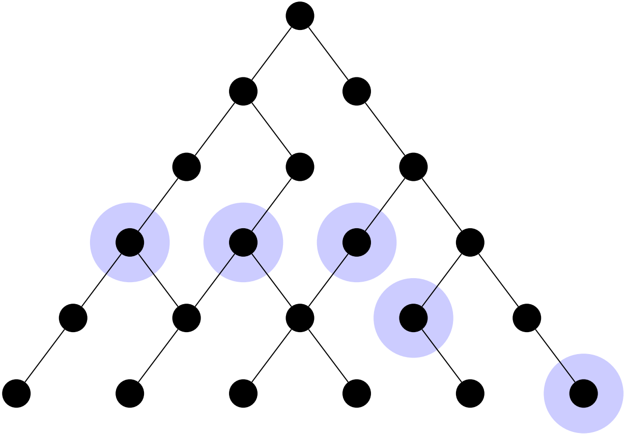
CTL in Pictures



EX ϕ

There *exists* a next state where ϕ holds.

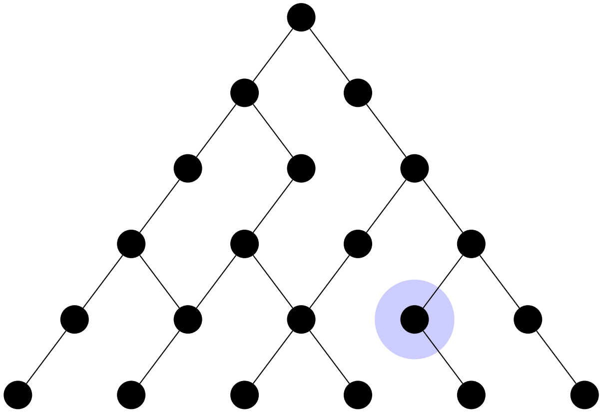
CTL in Pictures



AF ϕ

For all paths, there exists a future state where ϕ holds.

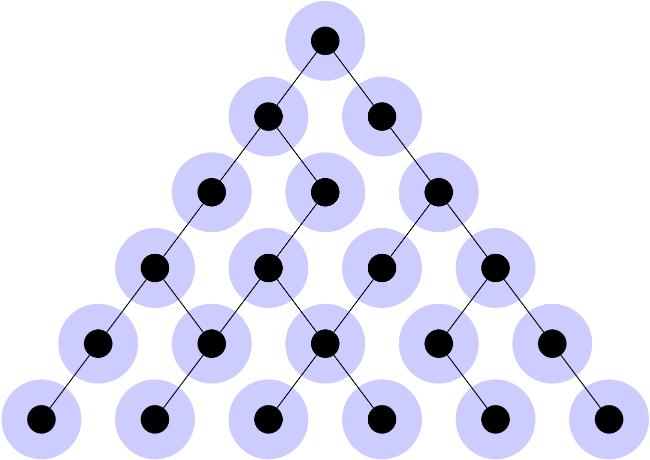
CTL in Pictures



EF ϕ

There exists a path with a future state where ϕ holds.

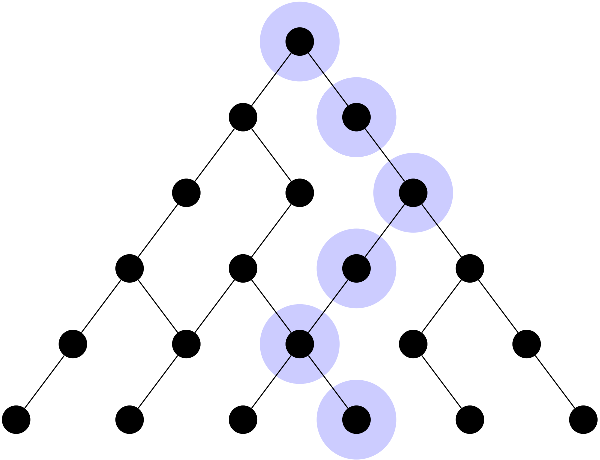
CTL in Pictures



AG ϕ

For all paths, for all states along them, ϕ holds.

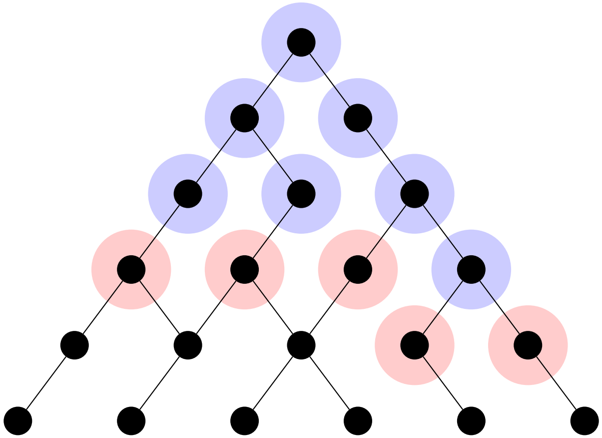
CTL in Pictures



EG ϕ

There exists a path such that, for all states along it, ϕ holds.

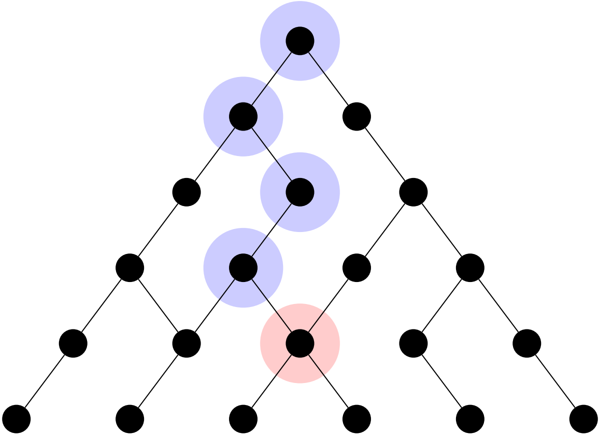
CTL in Pictures



$\mathbf{A}[\phi \; \mathbf{U} \; \psi]$

For all paths, ψ eventually holds, and ϕ holds at all states earlier.

CTL in Pictures



$\mathbf{E}[\phi \; \mathbf{U} \; \psi]$

There exists a path where ψ eventually holds, and ϕ holds at all states earlier.

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

- **AG AF** *enabled*
  A certain process is enabled infinitely often on every computation path

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

- **AG AF** *enabled*
  A certain process is enabled infinitely often on every computation path

- **AG** $(requested \rightarrow$ **AF** $acknowledged)$
  for any state, if a request ocurs, then it will eventually be acknowledged

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

- **AG AF** *enabled*
  A certain process is enabled infinitely often on every computation path

- **AG** $(\textit{requested} \rightarrow \textbf{AF}\ \textit{acknowledged})$
  for any state, if a request ocurs, then it will eventually be acknowledged

- **AG** $(\phi \rightarrow \mathbf{E}[\phi\ \mathbf{U}\ \psi])$
  for any state, if ϕ *holds, then there is a future where* ψ *eventually holds, and* ϕ *holds for all points in between*

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

- **AG AF** *enabled*
  A certain process is enabled infinitely often on every computation path

- **AG** $(\textit{requested} \rightarrow \textbf{AF}\ \textit{acknowledged})$
  for any state, if a request ocurs, then it will eventually be acknowledged

- **AG** $(\phi \rightarrow \mathbf{E}[\phi\ \mathbf{U}\ \psi])$
  for any state, if ϕ *holds, then there is a future where* ψ *eventually holds, and* ϕ *holds for all points in between*

- **AG** $(\phi \rightarrow \mathbf{EG}\ \psi)$
  for any state, if ϕ *holds then there is a future where* ψ *always holds*

Examples of CTL formulas (and their possible readings)

- **EF** ϕ
  it is possible to get to a state where ϕ *is true*

- **AG AF** *enabled*
  A certain process is enabled infinitely often on every computation path

- **AG** $($*requested* $\rightarrow$ **AF** *acknowledged*$)$
  for any state, if a request ocurs, then it will eventually be acknowledged

- **AG** $(\phi \rightarrow \mathbf{E}[\phi\ \mathbf{U}\ \psi])$
  for any state, if ϕ *holds, then there is a future where* ψ *eventually holds, and* ϕ *holds for all points in between*

- **AG** $(\phi \rightarrow \mathbf{EG}\ \psi)$
  for any state, if ϕ *holds then there is a future where* ψ *always holds*

- **EF AG** ϕ
  there exists a possible state in the future, from where ϕ *is always true*

CTL Equivalences

de Morgan dualities for the temporal connectives:

$$\begin{aligned} \neg\mathbf{EX}\,\phi &\equiv \mathbf{AX}\,\neg\phi \\ \neg\mathbf{EF}\,\phi &\equiv \mathbf{AG}\,\neg\phi \\ \neg\mathbf{EG}\,\phi &\equiv \mathbf{AF}\,\neg\phi \end{aligned}$$

Also have

$$\begin{aligned} \mathbf{AF}\,\phi &\equiv \mathbf{A}[\top\ \mathbf{U}\ \phi] \\ \mathbf{EF}\,\phi &\equiv \mathbf{E}[\top\ \mathbf{U}\ \phi] \\ \mathbf{A}[\phi\ \mathbf{U}\ \psi] &\equiv \neg(\mathbf{E}[\neg\psi\ \mathbf{U}\ (\neg\phi \wedge \neg\psi)] \vee \mathbf{EG}\,\neg\psi) \end{aligned}$$

From these, one can show that the sets $\{\mathbf{AU}, \mathbf{EU}, \mathbf{EX}\}$ and $\{\mathbf{EU}, \mathbf{EG}, \mathbf{EX}\}$ are both adequate sets of temporal connectives.

Differences between LTL and CTL

LTL allows for questions of the form

- For all paths, does the LTL formula ϕ hold?
- Does there exist a path on which the LTL formula ϕ holds?
  (Ask whether $\neg\phi$ holds on all paths, and ask for a counterexample)

CTL allows mixing of path quantifiers:

- $\mathbf{AG}\ (p \rightarrow \mathbf{EG}\ q)$
  For all paths, if p is true, then there exists a path on which q is always true.

However, some path properties are impossible to express in CTL

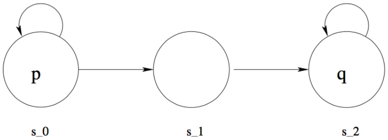
$$\left.\begin{array}{ll} \text{LTL:} & \mathbf{G\ F}\ p \rightarrow \mathbf{G\ F}\ q \\ \text{CTL:} & \mathbf{AG\ AF}\ p \rightarrow \mathbf{AG\ AF}\ q \end{array}\right\} \text{are not the same}$$

Exist *fair* refinements of CTL that address this issue to some extent.

- *E.g., path quantifiers that only consider paths where something happens infinitely often.*

LTL *vs* CTL

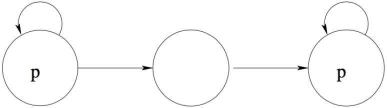
LTL: $\mathbf{G}\,\mathbf{F}\,p \rightarrow \mathbf{G}\,\mathbf{F}\,q$
CTL: $\mathbf{AG}\,\mathbf{AF}\,p \rightarrow \mathbf{AG}\,\mathbf{AF}\,q$
} are not the same



The CTL formula is trivially satisfied, because $\mathbf{AG}\,\mathbf{AF}\,p$ is not satisfied. The LTL formula is not satisfied, because the path cycling through s_0 forever satisfies $\mathbf{G}\,\mathbf{F}\,p$ but not $\mathbf{G}\,\mathbf{F}\,q$.

LTL *vs* CTL

LTL: $\mathbf{F}\,\mathbf{G}\,p$
CTL: $\mathbf{AF}\,\mathbf{AG}\,p$
} are not the same



Exercise: Why?

CTL Model Checking

CTL Model Checking seeks to answer the question: *is it the case that*

$$\mathcal{M}, s_0 \models \phi$$

for some initial state s_0?

CTL Model Checking algorithms usually fix $\mathcal{M} = \langle S, \rightarrow, L \rangle$ and ϕ and compute all states s of $\mathcal{M}$ that satisfy ϕ:

$$[\![\phi]\!]_{\mathcal{M}} = \{s \in S \mid \mathcal{M}, s \models \phi\}$$

"the denotation of ϕ in the model $\mathcal{M}$"

The model checking question now becomes: $s_0 \in [\![\phi]\!]_{\mathcal{M}}$?

(The model $\mathcal{M}$ is usually left implicit)

Denotation Semantics for CTL

We compute $[\![\phi]\!]$ recursively on the structure of ϕ:

$$\begin{array}{rcl} [\![\top]\!] & = & S \\ [\![\bot]\!] & = & \emptyset \\ [\![p]\!] & = & \{s \in S \mid p \in L(s)\} \\ [\![\neg\phi]\!] & = & S - [\![\phi]\!] \\ [\![\phi \wedge \psi]\!] & = & [\![\phi]\!] \cap [\![\psi]\!] \\ [\![\phi \vee \psi]\!] & = & [\![\phi]\!] \cup [\![\psi]\!] \\ [\![\phi \to \psi]\!] & = & (S - [\![\phi]\!]) \cup [\![\psi]\!] \end{array}$$

Since $[\![\phi]\!]$ is always a finite set, these are computable.

Denotation Semantics of the Temporal Connectives

$$\begin{aligned} [\![\mathbf{EX}\ \phi]\!] &= \mathrm{pre}_\exists([\![\phi]\!]) \\ [\![\mathbf{AX}\ \phi]\!] &= \mathrm{pre}_\forall([\![\phi]\!]) \end{aligned}$$

where

$$\begin{aligned} \mathrm{pre}_\exists(Y) &\doteq \{s \in S \mid \exists s' \in S.\ (s \to s') \land s' \in Y\} \\ \mathrm{pre}_\forall(Y) &\doteq \{s \in S \mid \forall s' \in S.\ (s \to s') \to s' \in Y\} \end{aligned}$$

these are again computable, because Y and S are finite.

But what about the rest of the temporal connectives? *e.g.*

$$[\![\mathbf{EF}\ \phi]\!] = \{s \in S \mid \exists \pi \text{ s.t. } s_0 = s.\ \exists i.\ s_i \models \phi\}$$

No obvious way to compute this: there are infinitely many paths π!

Approximating $[\![\mathbf{EF}\ \phi]\!]$

Define

$$\begin{aligned} \mathbf{EF}_0\ \phi &= \bot \\ \mathbf{EF}_{i+1}\ \phi &= \phi \vee \mathbf{EX}\ \mathbf{EF}_i\ \phi \end{aligned}$$

Then

$$\begin{aligned} \mathbf{EF}_1\ \phi &= \phi \\ \mathbf{EF}_2\ \phi &= \phi \vee \mathbf{EX}\ \phi \\ \mathbf{EF}_3\ \phi &= \phi \vee \mathbf{EX}\ (\phi \vee \mathbf{EX}\ \phi) \\ \dots \end{aligned}$$

$s \in [\![\mathbf{EF}_i\ \phi]\!]$ if there exists a finite path of length $i - 1$ from s and ϕ holds at some point along that path.

For a given (fixed) model $\mathbf{M}$, let $n = |S|$. If there is a path of length $k > n$ on which ϕ holds somewhere, there will also be a path of length n. *(Proof: take the k-length path and repeatedly cut out segments between repeated states.)*

Therefore, for all $k > n$, $[\![\mathbf{EF}_k\ \phi]\!] = [\![\mathbf{EF}_n\ \phi]\!]$

Computing $[\![\mathbf{EF}\ \phi]\!]$

By a similar argument,

$$[\![\mathbf{EF}\ \phi]\!] = [\![\mathbf{EF}_n\ \phi]\!]$$

The approximations can be computed by recursion on *i*:

$$\begin{array}{rcl} [\![\mathbf{EF}_0\ \phi]\!] & = & \emptyset \\ [\![\mathbf{EF}_{i+1}\ \phi]\!] & = & [\![\phi]\!] \cup \mathrm{pre}_{\exists}([\![\mathbf{EF}_i\ \phi]\!]) \end{array}$$

So we have an effective way of computing $[\![\mathbf{EF}\ \phi]\!]$.

Approximating $[\![\mathbf{EG}\ \phi]\!]$

Define

$$\begin{aligned} \mathbf{EG}_0\ \phi &= \top \\ \mathbf{EG}_{i+1}\ \phi &= \phi \wedge \mathbf{EX}\ \mathbf{EG}_i\ \phi \end{aligned}$$

Then

$$\begin{aligned} \mathbf{EG}_1\ \phi &= \phi \\ \mathbf{EG}_2\ \phi &= \phi \wedge \mathbf{EX}\ \phi \\ \mathbf{EG}_3\ \phi &= \phi \wedge \mathbf{EX}\ (\phi \wedge \mathbf{EX}\ \phi) \\ \ldots \end{aligned}$$

$s \in [\![\mathbf{EG}_i\ \phi]\!]$ if there exists a finite path of length $i - 1$ from s and ϕ holds at every point along that path.

As with $[\![\mathbf{EF}\ \phi]\!]$, we have for all $k > n$, $[\![\mathbf{EG}_k\ \phi]\!] = [\![\mathbf{EG}_n\ \phi]\!] = [\![\mathbf{EG}\ \phi]\!]$ and so we can compute $[\![\mathbf{EG}\ \phi]\!]$.

Fixed point Theory

What's happening here is that we are computing fixed points.

A set $X \subseteq S$ is a *fixed point* of a function $F : \mathcal{P}(S) \to \mathcal{P}(S)$ iff $F(X) = X$.

We have that (for $n = |S|$)

$$
\begin{array}{rcl}
[\![\mathbf{EF}_n\ \phi]\!] & = & [\![\mathbf{EF}_{n+1}\ \phi]\!] \\
 & = & [\![\phi \vee \mathbf{EX}\ \mathbf{EF}_n\ \phi]\!] \\
 & = & [\![\phi]\!] \cup \mathrm{pre}_\exists([\![\mathbf{EF}_n\ \phi]\!])
\end{array}
$$

so $[\![\mathbf{EF}_n]\!]$ is a fixed point of $F(Y) = [\![\phi]\!] \cup \mathrm{pre}_\exists(Y)$.

Also, $[\![\mathbf{EF}\ \phi]\!]$ is a fixed point of F, since $[\![\mathbf{EF}\ \phi]\!] = [\![\mathbf{EF}_n\ \phi]\!]$.

More specifically, they are both the least fixed point of F.

Fixed point Theorem

Let $F : \mathcal{P}(S) \to \mathcal{P}(S)$ be a function that takes sets to sets.

- F is monotone iff $X \subseteq Y$ implies $F(X) \subseteq F(Y)$.
- Let $F^0(X) = X$ and $F^{i+1}(X) = F(F^i(X))$.
- Given a collection of sets $C \subseteq \mathcal{P}(S)$, a set $X \in C$ is
  1. the least element of C if $\forall Y \in C.\ X \subseteq Y$; and
  2. the greatest element of C if $\forall Y \in C.\ Y \subseteq X$.

Theorem (Knaster-Tarski (Special Case))

Let S be a set with n elements and $F : \mathcal{P}(S) \to \mathcal{P}(S)$ be a monotone function. Then

- $F^n(\emptyset)$ *is the* least *fixed point of F; and*
- $F^n(S)$ *is the* greatest *fixed point of F.*

(Proof: see H&R, Section 3.7.1)

This theorem justifies $F^n(\emptyset)$ and $F^n(S)$ being fixed points of F without the need, as before, to appeal to further details about F.

Denotational semantics of temporal connectives

When $F : \mathcal{P}(S) \to \mathcal{P}(S)$ is a monotone function, we write

- $\mu Y.\ F(Y)$ for the least fixed point of F; and
- $\nu Y.\ F(Y)$ for the greatest fixed point of F.

With this notation, we can define:

$$
\begin{array}{lcl}
[\![\mathbf{EF}\ \phi]\!] & = & \mu Y.\ [\![\phi]\!] \cup \mathrm{pre}_{\exists}(Y) \\
[\![\mathbf{EG}\ \phi]\!] & = & \nu Y.\ [\![\phi]\!] \cap \mathrm{pre}_{\exists}(Y) \\
[\![\mathbf{AF}\ \phi]\!] & = & \mu Y.\ [\![\phi]\!] \cup \mathrm{pre}_{\forall}(Y) \\
[\![\mathbf{AG}\ \phi]\!] & = & \nu Y.\ [\![\phi]\!] \cap \mathrm{pre}_{\forall}(Y) \\
[\![\mathbf{E}[\phi\ \mathbf{U}\ \psi]]\!] & = & \mu Y.\ [\![\psi]\!] \cup ([\![\phi]\!] \cap \mathrm{pre}_{\exists}(Y)) \\
[\![\mathbf{A}[\phi\ \mathbf{U}\ \psi]]\!] & = & \mu Y.\ [\![\psi]\!] \cup ([\![\phi]\!] \cap \mathrm{pre}_{\forall}(Y))
\end{array}
$$

In every case, F is monotone, so the Knaster-Tarski theorem assures us that the fixed point exists, and can be computed.

Further CTL Equivalences

The fixed point characterisations of the CTL temporal connectives justify some more equivalences between CTL formulas:

$$
\begin{array}{lcl}
\mathbf{EF}\,\phi & \equiv & \phi \vee \mathbf{EX}\,\mathbf{EF}\,\phi \\
\mathbf{EG}\,\phi & \equiv & \phi \wedge \mathbf{EX}\,\mathbf{EG}\,\phi \\
\mathbf{AF}\,\phi & \equiv & \phi \vee \mathbf{AX}\,\mathbf{AF}\,\phi \\
\mathbf{AG}\,\phi & \equiv & \phi \wedge \mathbf{AX}\,\mathbf{AG}\,\phi \\
\mathbf{E}[\phi\;\mathbf{U}\;\psi] & \equiv & \psi \vee (\phi \wedge \mathbf{EX}\,\mathbf{E}[\phi\;\mathbf{U}\;\psi]) \\
\mathbf{A}[\phi\;\mathbf{U}\;\psi] & \equiv & \psi \vee (\phi \wedge \mathbf{AX}\,\mathbf{A}[\phi\;\mathbf{U}\;\psi])
\end{array}
$$

Summary

- CTL (H&R 3.4, 3.5, 3.6.1, 3.7)
  - CTL, Syntax and Semantics
  - Comparison with LTL
  - Model Checking algorithm for CTL
- Next time:
  - (A taste of) The LTL Model Checking algorithm