Fairness (revisited)

Fairness definition

For $TS = (S, Act, \rightarrow, I, AP, L)$ without terminal states, $A \subseteq Act$, and infinite execution fragment $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ of TS:

1. ρ is unconditionally A-fair whenever:
   $\text{true} \implies \underbrace{\forall k \geq 0.\, \exists j \geq k.\, \alpha_j \in A}_{\text{infinitely often } A \text{ is taken}}$
2. ρ is strongly A-fair whenever:
   $\underbrace{(\forall k \geq 0.\, \exists j \geq k.\, Act(s_j) \cap A \neq \emptyset)}_{\text{infinitely often } A \text{ is enabled}} \implies \underbrace{(\forall k \geq 0.\, \exists j \geq k.\, \alpha_j \in A)}_{\text{infinitely often } A \text{ is taken}}$
3. ρ is weakly A-fair whenever:
   $\underbrace{(\exists k \geq 0.\, \forall j \geq k.\, Act(s_j) \cap A \neq \emptyset)}_{A \text{ is eventually always enabled}} \implies \underbrace{(\forall k \geq 0.\, \exists j \geq k.\, \alpha_j \in A)}_{\text{infinitely often } A \text{ is taken}}$

where $Act(s) = \{\alpha \in Act \mid \exists s' \in S.\, s \xrightarrow{\alpha} s'\}$

Fair satisfaction

- TS satisfies LT-property P:

  $TS \models P$ if and only if $Traces(TS) \subseteq P$

  - TS satisfies the LT property P if all its observable behaviors are admissible
- TS fairly satisfies LT-property P wrt. fairness assumption $\mathcal{F}$:

  $TS \models_{\mathcal{F}} P$ if and only if $FairTraces_{\mathcal{F}}(TS) \subseteq P$

  - if all paths in TS are $\mathcal{F}$-fair, then $TS \models_{\mathcal{F}} P$ if and only if $TS \models P$
  - if some path in TS is not $\mathcal{F}$-fair, then possibly $TS \models_{\mathcal{F}} P$ but $TS \not\models P$

LTL fairness constraints

Let Φ and Ψ be propositional logic formulas over AP.

1. An unconditional LTL fairness constraint is of the form:

   $$ufair = \Box\Diamond\Psi$$

2. A strong LTL fairness condition is of the form:

   $$sfair = \Box\Diamond\Phi \longrightarrow \Box\Diamond\Psi$$

3. A weak LTL fairness constraint is of the form:

   $$wfair = \Diamond\Box\Phi \longrightarrow \Box\Diamond\Psi$$

Φ stands for “something is enabled”; Ψ for “something is taken”

Fair satisfaction

LTL fairness assumption = conjunction of LTL fairness constraints: $fair = ufair \wedge sfair \wedge wfair$

For state s in transition system TS (over AP) without terminal states, let

$$FairPaths_{fair}(s) = \{ \pi \in Paths(s) \mid \pi \models fair \}$$
$$FairTraces_{fair}(s) = \{ trace(\pi) \mid \pi \in FairPaths_{fair}(s) \}$$

For LTL-formula φ, and LTL fairness assumption $fair$:

$s \models_{fair} \varphi$ if and only if $\forall \pi \in FairPaths_{fair}(s).\ \pi \models \varphi$ and

$TS \models_{fair} \varphi$ if and only if $\forall s_0 \in I.\ s_0 \models_{fair} \varphi$

$\models_{fair}$ is the **fair satisfaction relation** for LTL;
$\models$ the standard one for LTL

Turning action-based into state-based fairness

For $TS = (S, Act, \rightarrow, I, AP, L)$ let
$TS' = (S', Act \cup \{ begin \}, \rightarrow', I', AP', L')$ with:

- $S' = I \times \{ begin \} \cup S \times Act$ and $I' = I \times \{ begin \}$
- $\rightarrow'$ is the smallest relation satisfying:

$$\frac{s \xrightarrow{\alpha} s'}{\langle s, \beta \rangle \xrightarrow{\alpha}' \langle s', \alpha \rangle} \quad \text{and} \quad \frac{s_0 \xrightarrow{\alpha} s \ \ s_0 \in I}{\langle s_0, begin \rangle \xrightarrow{\alpha}' \langle s, \alpha \rangle}$$

- $AP' = AP \cup \{ enabled(\alpha), taken(\alpha) \mid \alpha \in Act \}$
- labeling function:
  - $L'(\langle s_0, begin \rangle) = L(s_0) \cup \{ enabled(\beta) \mid \beta \in Act(s_0) \}$
  - $L'(\langle s, \alpha \rangle) = L(s) \cup \{ taken(\alpha) \} \cup \{ enabled(\beta) \mid \beta \in Act(s) \}$

it follows: $Traces_{AP}(TS) = Traces_{AP}(TS')$

State- versus action-based fairness

- Strong A-fairness is described by the LTL fairness assumption:

$$sfair_A = \mathsf{GF} \bigvee_{\alpha \in A} enabled(\alpha) \rightarrow \mathsf{GF} \bigvee_{\alpha \in A} taken(\alpha)$$

- The fair traces of TS and its action-based variant TS' are equal:

$$\{ trace_{AP}(\pi) \mid \pi \in Paths(TS), \pi \text{ is } \mathcal{F}\text{-fair} \}$$
$$= \{ trace_{AP}(\pi') \mid \pi' \in Paths(TS'), \pi' \models fair \}$$

- For every LT-property P (over AP):
  $TS \models_{\mathcal{F}} P$ iff $TS' \models_{fair} P$

Reducing $\models_{fair}$ to $\models$

For:

- transition system TS without terminal states
- LTL formula φ, and
- LTL fairness assumption $fair$

it holds:

$$TS \models_{fair} \varphi \quad \text{if and only if} \quad TS \models (fair \rightarrow \varphi)$$

verifying an LTL-formula under a fairness assumption can be done using standard verification algorithms for LTL

Fairness constraints in CTL

- For LTL it holds:
  $TS \models_{fair} \varphi$ if and only if $TS \models (fair \rightarrow \varphi)$
- An analogous approach for CTL is not possible!
- Formulas of form $\forall(fair \rightarrow \varphi)$ and $\exists(fair \wedge \varphi)$ needed
- But: boolean combinations of path formulas are not allowed in CTL
- and: strong fairness constraints

$$\mathsf{GF}\, b \rightarrow \mathsf{GF}\, c \equiv \mathsf{FG} \neg b \vee \mathsf{GF}\, c$$

  cannot be expressed, since persistence properties are not in CTL
- Solution: change the semantics of CTL by ignoring unfair paths

CTL fairness constraints

- A **strong** CTL fairness constraint is a formula of the form:

$$sfair = \bigwedge_{0<i\leq k} (\mathsf{GF}\, \Phi_i \rightarrow \mathsf{GF}\, \Psi_i)$$

  - where Φ_i and Ψ_i (for $0 < i \leq k$) are CTL-formulas over AP
- **unconditional** and **weak** CTL fairness constraints are defined analogously,

$$ufair = \bigwedge_{0<i\leq k} \mathsf{GF}\, \Psi_i \quad \text{and} \quad wfair = \bigwedge_{0<i\leq k} (\mathsf{FG}\, \Phi_i \rightarrow \mathsf{GF}\, \Psi_i)$$

$\Rightarrow$ a CTL fairness constraint is an LTL formula over CTL state formulas!

- a **CTL fairness assumption** $fair$ is a conjunction of CTL fairness constraints.

Semantics of fair CTL

For CTL fairness assumption $fair$, relation $\models_{fair}$ is defined by:

$s \models_{fair} a$ iff $a \in Label(s)$
$s \models_{fair} \neg \Phi$ iff $\neg (s \models_{fair} \Phi)$
$s \models_{fair} \Phi \vee \Psi$ iff $(s \models_{fair} \Phi) \vee (s \models_{fair} \Psi)$
$s \models_{fair} \mathsf{E}\varphi$ iff $\pi \models_{fair} \varphi$ for some fair path π that starts in s
$s \models_{fair} \mathsf{A}\varphi$ iff $\pi \models_{fair} \varphi$ for all fair paths π that start in s

$\pi \models_{fair} \mathsf{X}\Phi$ iff $\pi[1] \models_{fair} \Phi$
$\pi \models_{fair} \Phi \mathsf{U} \Psi$ iff $(\exists j \geq 0.\, \pi[j] \models_{fair} \Psi \wedge (\forall\, 0 \leq k < j.\, \pi[k] \models_{fair} \Phi))$

π is a fair path iff $\pi \models fair$ for CTL fairness assumption $fair$

Transition system semantics

- For CTL state formula Φ, and fairness assumption $fair$, the **satisfaction set** $Sat_{fair}(\Phi)$ is defined by:

$$Sat_{fair}(\Phi) = \{\, q \in Q \mid q \models_{fair} \Phi \,\}$$

- TS satisfies CTL-formula Φ iff Φ holds in all its initial states:

$$TS \models_{fair} \Phi \quad \text{if and only if} \quad \forall q_0 \in I.\, q_0 \models_{fair} \Phi$$

  - this is equivalent to $I \subseteq Sat_{fair}(\Phi)$

Fair CTL model-checking problem

For:

- finite transition system
- CTL formula Φ in ENF, and
- CTL fairness assumption *fair*

establish whether or not:

$$TS \models_{fair} \Phi$$

use bottom-up procedure a la CTL to determine $Sat_{fair}(\Phi)$
using as much as possible standard CTL model-checking algorithms

CTL fairness constraints

- A **strong CTL fairness constraint**:
  $sfair = \bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\,\Phi_i \rightarrow \mathsf{G}\,\mathsf{F}\,\Psi_i)$
  - where Φ_i and Ψ_i (for $0 < i \leq k$) are CTL-formulas over AP
- Replace the CTL state formulas in *sfair* by fresh atomic propositions:

$$sfair := \bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\,a_i \rightarrow \mathsf{G}\,\mathsf{F}\,b_i)$$

  - where $a_i \in L(s)$ if and only if $s \in Sat(\Phi_i)$ (not $Sat_{fair}(\Phi_i)$!)
  - $\ldots$ $b_i \in L(s)$ if and only if $s \in Sat(\Psi_i)$ (not $Sat_{fair}(\Psi_i)$!)
  - (for unconditional and weak fairness this goes similarly)
- Note:
  $\pi \models fair$ iff $\pi[j..] \models fair$ for some $j \geq 0$ iff $\pi[j..] \models fair$ for all $j \geq 0$

Results for $\models_{fair}$ (1)

$s \models_{fair} \mathsf{EX}\,a$ if and only if $\exists s' \in Post(s)$
with $s' \models a$ and $FairPaths(s') \neq \emptyset$

$s \models_{fair} \mathsf{E}(a\,\mathsf{U}\,a')$ if and only if there exists a finite path fragment

$$s_0\,s_1\,s_2 \ldots s_{n-1}\,s_n \in Paths_{fin}(s) \quad \text{with } n \geq 0$$

such that $s_i \models a$ for $0 \leq i < n$, $s_n \models a'$, and $FairPaths(s_n) \neq \emptyset$

Results for $\models_{fair}$ (2)

$s \models_{fair} \mathsf{EX}\,a$ if and only if $\exists s' \in Post(s)$
with $s' \models a$ and $\underbrace{FairPaths(s') \neq \emptyset}_{s' \models_{fair} \mathsf{EG}\,\mathsf{true}}$

$s \models_{fair} \mathsf{E}(a\,\mathsf{U}\,a')$ if and only if there exists a finite path fragment

$$s_0\,s_1\,s_2 \ldots s_{n-1}\,s_n \in Paths_{fin}(s) \quad \text{with } n \geq 0$$

such that $s_i \models a$ for $0 \leq i < n$, $s_n \models a'$, and $\underbrace{FairPaths(s_n) \neq \emptyset}_{s_n \models_{fair} \mathsf{EG}\,\mathsf{true}}$

Basic algorithm

- Determine $Sat_{fair}(\mathrm{EG}\,\mathrm{true}) = \{ q \in Q \mid FairPaths(q) \neq \emptyset \}$
- Introduce an atomic proposition a_{fair} such that:
  - $a_{fair} \in L(q)$ if and only if $q \in Sat_{fair}(\mathrm{EG}\,\mathrm{true})$
- Compute the sets $Sat_{fair}(\Psi)$ for all subformulas Ψ of Φ (in ENF) by:

$$\begin{array}{rcl} Sat_{fair}(a) & = & \{ q \in Q \mid a \in L(q) \} \\ Sat_{fair}(\neg a) & = & Q \setminus Sat_{fair}(a) \\ Sat_{fair}(a \wedge a') & = & Sat_{fair}(a) \cap Sat_{fair}(a') \\ Sat_{fair}(\mathrm{EX}\,a) & = & Sat(\mathrm{EX}\,(a \wedge a_{fair})) \\ Sat_{fair}(\mathrm{E}\,(a\,\mathrm{U}\,a')) & = & Sat(\mathrm{E}\,(a\,\mathrm{U}\,(a' \wedge a_{fair}))) \\ Sat_{fair}(\mathrm{EG}\,a) & = & \ldots\ldots \end{array}$$

- Thus: model checking CTL under fairness constraints is
  - CTL model checking + algorithm for computing $Sat_{fair}(\mathrm{EG}\,a)$!

Core model-checking algorithm

{states are assumed to be labeled with a_i and b_i}
compute $Sat_{fair}(\mathrm{EG}\,\mathrm{true}) = \{ q \in Q \mid FairPaths(q) \neq \emptyset \}$
forall $q \in Sat_{fair}(\mathrm{EG}\,\mathrm{true})$ **do** $L(q) := L(q) \cup \{ a_{fair} \}$ **od**
{compute $Sat_{fair}(\Phi)$}
for all $0 < i \leq |\Phi|$ **do**
  for all $\Psi \in Sub(\Phi)$ with $|\Psi| = i$ **do**
    switch(Ψ):

true	:	$Sat_{fair}(\Psi) := Q$;
a	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a \in L(s) \}$;
$a \wedge a'$	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a, a' \in L(s) \}$;
$\neg a$	:	$Sat_{fair}(\Psi) := \{ q \in Q \mid a \notin L(s) \}$;
$\mathrm{EX}\,a$	:	$Sat_{fair}(\Psi) := Sat(\mathrm{EX}\,(a \wedge a_{fair}))$;
$\mathrm{E}\,(a\,\mathrm{U}\,a')$	:	$Sat_{fair}(\Psi) := Sat(\mathrm{E}\,(a\,\mathrm{U}\,(a' \wedge a_{fair})))$;
$\mathrm{EG}\,a$	:	compute $Sat_{fair}(\mathrm{EG}\,a)$

    end switch
    replace all occurrences of Ψ (in Φ) by the fresh atomic proposition a_Ψ
    forall $q \in Sat_{fair}(\Psi)$ **do** $L(q) := L(q) \cup \{ a_\Psi \}$ **od**
  end for
end for
return $I \subseteq Sat_{fair}(\Phi)$

Characterization of $Sat_{fair}(\mathrm{EG}\,a)$

$$q \models_{sfair} \mathrm{EG}\,a \quad \text{where} \quad sfair = \bigwedge_{0<i\leq k} (\mathrm{G}\,\mathrm{F}\,b_i \rightarrow \mathrm{G}\,\mathrm{F}\,c_i)$$

iff there exists a finite path fragment $q_0 \ldots q_n$ and a cycle $q'_0 \ldots q'_r$ with:

1. $q_0 = q$ and $q_n = q'_0 = q'_r$
2. $q_i \models a$, for every $0 \leq i \leq n$, and $q'_j \models a$, for every $0 \leq j \leq r$, and
3. $Sat(b_i) \cap \{ q'_1, \ldots, q'_r \} = \emptyset$ or $Sat(c_i) \cap \{ q'_1, \ldots, q'_r \} \neq \emptyset$ for $0 < i \leq k$

Computing $Sat_{fair}(\mathrm{EG}\,a)$

- Consider state q only if $q \models a$, otherwise eliminate q
  - change TS into $TS[a] = (S', Act, \rightarrow', I', AP, L')$ with $S' = Sat(a)$,
  - $\rightarrow' = \rightarrow \cap (S' \times Act \times S')$, $I' = I \cap S'$, and $L'(s) = L(s)$ for $s \in S'$

  $\Rightarrow$ each infinite path fragment in $TS[a]$ satisfies $\mathrm{G}\,a$
- $q \models_{fair} \mathrm{EG}\,a$ iff there is a non-trivial SCC D in $TS[a]$ reachable from q such that
  - $D \cap Sat(b_i) = \emptyset$ or
  - $D \cap Sat(c_i) \neq \emptyset$

  for $0 < i \leq k$
- $Sat_{sfair}(\mathrm{EG}\,a) = \{ q \in S \mid Reach_{TS[a]}(s) \cap T \neq \emptyset \}$
  - T is the union of all such SCCs D.

how to compute T?

Unconditional fairness

$$ufair \equiv \bigwedge_{0<i\leq k} \mathsf{GF}\, b_i$$

Let T be the set union of all non-trivial SCCs C of $TS[a]$ satisfying

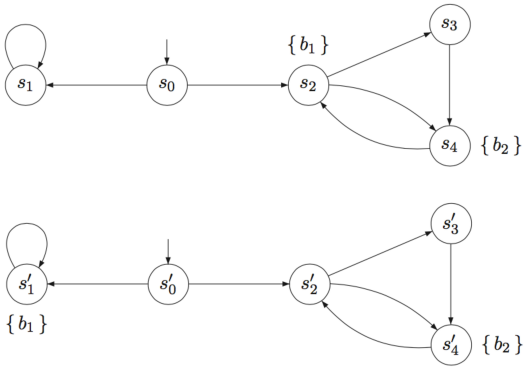
$$C \cap Sat(b_i) \neq \emptyset \quad \text{for all } 0 < i \leq k$$

It now follows:

$$s \models_{ufair} \mathsf{EG}\, a \quad \text{if and only if} \quad Reach_{TS[a]}(s) \cap T \neq \emptyset$$

$\Rightarrow$ T can be determined by a simple graph analysis (DFS)

Examples $\mathsf{GF}\, b_1 \wedge \mathsf{GF}\, b_2$



Weak fairness

- Weak fairness constraint $\bigwedge_{0<i\leq k} (\mathsf{FG}\, b_i \rightarrow \mathsf{GF}\, c_i)$
  $= \bigwedge_{0<i\leq k} (\mathsf{GF}\, (\neg b_i) \vee \mathsf{GF}\, c_i)$
  $= \bigwedge_{0<i\leq k} (\mathsf{GF}\, (\neg b_i \vee c_i))$
- can be treated like an unconditional fairness constraint.

Strong fairness: single constraint ($k = 1$)

- $sfair = \mathsf{GF}\, b_1 \rightarrow \mathsf{GF}\, c_1$
- $q \models_{sfair} \mathsf{EG}\, a$ iff C is a non-trivial SCC in $TS[a]$ reachable from q with:
  (1) $C \cap Sat(c_1) \neq \emptyset$, or
  (2) there exists a non-trivial SCC D in $C[\neg b_1]$
- For the union T of all such SCCs C:

$$q \models_{sfair} \mathsf{EG}\, a \quad \text{if and only if} \quad Reach_{TS[a]}(q) \cap T \neq \emptyset$$

Strong fairness: general case ($k > 1$)

Check each non-trivial SCC C recursively as follows:
Check(C, $\bigwedge_{0<i\leq k} (\mathsf{G}\,\mathsf{F}\,b_i \rightarrow \mathsf{G}\,\mathsf{F}\,c_i)$):

if $\forall i \in \{1,\ldots,k\} : C \cap \mathit{Sat}(c_i) \neq \emptyset$ **return** *true*
else
  choose some $j \in \{1,\ldots,k\} : C \cap \mathit{Sat}(c_j) = \emptyset$.
  remove all states in $\mathit{Sat}(b_j)$ from C
  for all non-trivial SCCs D do
    if Check(D, $\bigwedge_{0<i\leq k, i\neq j} (\mathsf{G}\,\mathsf{F}\,b_i \rightarrow \mathsf{G}\,\mathsf{F}\,c_i)$) **return** *true*
  return *false*

T is the union of all SCCs C that pass the check.

CTL*

Time complexity

For transition system TS with N states and M edges,
CTL formula Φ, and CTL fairness constraint fair with k conjuncts,
the CTL model-checking problem $TS \models_{\mathit{fair}} \Phi$
can be determined in time $\mathcal{O}(|\,\Phi\,|\cdot(N+M)\cdot k)$

Syntax of CTL*

CTL* state formulas are formed according to the grammar:

$$\Phi ::= \mathsf{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathsf{E}\,\varphi$$

where $a \in \mathit{AP}$ and φ is a path-formula

CTL* path formulas are formed according to the grammar:

$$\varphi ::= \Phi \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \mathsf{X}\,\varphi \mid \varphi_1\,\mathsf{U}\,\varphi_2$$

where Φ is a state-formula, and φ, φ_1 and φ_2 are path-formulas

in CTL*: $\mathsf{A}\,\varphi = \neg\mathsf{E}\neg\varphi$.

CTL* semantics

$$
\begin{array}{lll}
s \models a & \text{iff} & a \in L(s) \\
s \models \neg \Phi & \text{iff} & \text{not } s \models \Phi \\
s \models \Phi \wedge \Psi & \text{iff} & (s \models \Phi) \text{ and } (s \models \Psi) \\
s \models \mathsf{E}\,\varphi & \text{iff} & \pi \models \varphi \text{ for some } \pi \in \mathit{Paths}(s)
\end{array}
$$

$$
\begin{array}{lll}
\pi \models \Phi & \text{iff} & \pi[0] \models \Phi \\
\pi \models \varphi_1 \wedge \varphi_2 & \text{iff} & \pi \models \varphi_1 \text{ and } \pi \models \varphi_2 \\
\pi \models \neg\varphi & \text{iff} & \text{not } \pi \models \varphi \\
\pi \models \mathsf{X}\,\Phi & \text{iff} & \pi[1..] \models \Phi \\
\pi \models \Phi\,\mathsf{U}\,\Psi & \text{iff} & \exists j \geq 0.\ (\pi[j..] \models \Psi \ \wedge\ (\forall\, 0 \leq k < j.\ \pi[k..] \models \Phi))
\end{array}
$$

Transition system semantics

- For CTL*-state-formula Φ, the **satisfaction set** $\mathit{Sat}(\Phi)$ is defined by:

$$\mathit{Sat}(\Phi) \ = \ \{\, q \in S \mid q \models \Phi \,\}$$

- TS satisfies CTL*-formula Φ iff Φ holds in all its initial states:

$$\mathit{TS} \models \Phi \quad \text{if and only if} \quad \forall q \in I.\ q_0 \models \Phi$$

this is exactly as for CTL

Embedding of LTL in CTL*

For LTL formula φ and TS without terminal states (both over AP) and for each $q \in S$:

$$\underbrace{q \models \varphi}_{\text{LTL semantics}} \quad \text{if and only if} \quad \underbrace{q \models \mathsf{A}\,\varphi}_{\text{CTL* semantics}}$$

Hence also:

$$\mathit{TS} \models_{\mathit{LTL}} \varphi \quad \text{if and only if} \quad \mathit{TS} \models_{\mathit{CTL*}} \mathsf{A}\,\varphi$$

CTL* is more expressive than LTL and CTL

For the CTL*-formula over $\mathit{AP} = \{\, a, b \,\}$:

$$\Phi \ = \ (\mathsf{AF}\,\mathsf{G}\ a) \ \vee \ (\mathsf{AG}\,\mathsf{EF}\ b)$$

there does **not** exist any equivalent LTL or CTL formula

CTL$^+$

CTL$^+$ state formulas are formed according to the grammar:

$$\Phi ::= \text{true} \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid \mathsf{E}\,\varphi \mid \mathsf{A}\,\varphi$$

where $a \in AP$ and φ is a path-formula

CTL$^+$ path formulas are formed according to the grammar:

$$\varphi ::= \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \mathsf{X}\,\Phi \mid \Phi_1\,\mathsf{U}\,\Phi_2$$

where Φ, Φ_1, Φ_2 are state-formulas, and φ, φ_1 and φ_2 are path-formulas

CTL$^+$ is as expressive as CTL

For example: $\underbrace{\mathsf{E}(\mathsf{F}\,a \wedge \mathsf{F}\,b)}_{\text{CTL}^+\text{ formula}} \equiv \underbrace{\mathsf{E}\mathsf{F}(a \wedge \mathsf{E}\mathsf{F}\,b) \vee \mathsf{E}\mathsf{F}(b \wedge \mathsf{E}\mathsf{F}\,a)}_{\text{CTL formula}}$

Some rules for transforming CTL$^+$ formulas into equivalent CTL formulas:

$$\begin{aligned}
\mathsf{E}(\neg(\Phi_1\,\mathsf{U}\,\Phi_2)) &\equiv \mathsf{E}((\Phi_1 \wedge \neg\Phi_2)\,\mathsf{U}\,(\neg\Phi_1 \wedge \neg\Phi_2)) \vee \mathsf{E}\mathsf{G}\neg\Phi_2 \\
\mathsf{E}(\mathsf{X}\Phi_1 \wedge \mathsf{X}\Phi_2) &\equiv \mathsf{E}\mathsf{X}(\Phi_1 \wedge \Phi_2) \\
\mathsf{E}(\mathsf{X}\Phi \wedge (\Phi_1\,\mathsf{U}\,\Phi_2)) &\equiv (\Phi_2 \wedge \mathsf{E}\mathsf{X}\Phi) \vee (\Phi_1 \wedge \mathsf{E}\mathsf{X}(\Phi \wedge \mathsf{E}(\Phi_1\,\mathsf{U}\,\Phi_2))) \\
\mathsf{E}((\Phi_1\,\mathsf{U}\,\Phi_2) \wedge (\Psi_1\,\mathsf{U}\,\Psi_2)) &\equiv \mathsf{E}((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Phi_2 \wedge \mathsf{E}(\Psi_1\,\mathsf{U}\,\Psi_2))) \vee \\
&\quad\ \mathsf{E}((\Phi_1 \wedge \Psi_1)\,\mathsf{U}\,(\Psi_2 \wedge \mathsf{E}(\Phi_1\,\mathsf{U}\,\Phi_2))) \\
&\ \ \vdots
\end{aligned}$$

adding boolean combinations of path formulas to CTL does not change its expressiveness but CTL$^+$ formulas can be much shorter than shortest equivalent CTL formulas

CTL* model checking

- Adopt the same bottom-up procedure as for (fair) CTL
- Replace each maximal proper state subformula Ψ by new proposition a_Ψ
  - $a_\Psi \in L(s)$ if and only if $s \in Sat(\Psi)$
- Most interesting case: formulas of the form $\mathsf{E}\,\varphi$
  - by replacing all maximal state sub-formulas in φ, an LTL-formula results!
- $q \models \mathsf{E}\,\varphi$ iff $\underbrace{q \not\models \mathsf{A}\neg\varphi}_{\text{CTL}^*\text{ semantics}}$ iff $\underbrace{q \not\models \neg\varphi}_{\text{LTL semantics}}$
  - $Sat_{CTL^*}(\mathsf{E}\,\varphi) = S \setminus Sat_{LTL}(\neg\varphi)$

CTL* model-checking algorithm

for all $i \leq |\,\Phi\,|$ **do**
  for all $\Psi \in Sub(\Phi)$ with $|\,\Psi\,| = i$ **do**
    switch(Ψ):

true	:	$Sat(\Psi) := S$;
a	:	$Sat(\Psi) := \{\, q \in S \mid a \in L(q)\,\}$;
$a_1 \wedge a_2$	:	$Sat(\Psi) := Sat(a_1) \cap Sat(a_2)$;
$\neg a$	:	$Sat(\Psi) := S \setminus Sat(a)$;
$\mathsf{E}\,\varphi$	:	determine $Sat_{LTL}(\neg\varphi)$ with an LTL model checker;
	:	$Sat(\Psi) := S \setminus Sat_{LTL}(\neg\varphi)$

    end switch
    $AP := AP \cup \{\, a_\Psi \,\}$; {introduce fresh atomic proposition}
    replace Ψ with a_Ψ
    forall $q \in Sat(\Psi)$ **do** $L(q) := L(q) \cup \{\, a_\Psi \,\}$; **od**
  end for
end for
return $I \subseteq Sat(\Phi)$

Time complexity

For transition system TS with N states and M transitions, CTL* formula Φ, the CTL* model-checking problem $TS \models \Phi$ can be determined in time $\mathcal{O}((N+M)\cdot 2^{|\Phi|})$.

the CTL* model-checking problem is PSPACE-complete