Outline

Introduction

Models

Specifications

Satisfiability and Model Checking for LTL

5 Branching Time Specifications
- CTL*
- CTL
- Fair CTL

Possibility is not expressible in LTL

Example:

φ: Whenever p holds, it is possible to reach a state where q holds.
φ cannot be expressed in LTL.

We need quantifications on runs: $\varphi = \mathsf{AG}(p \to \mathsf{EF}\, q)$

- E: for some infinite run
- A: for all infinite runs

Outline

Introduction

Models

Specifications

Satisfiability and Model Checking for LTL

5 Branching Time Specifications
- CTL*
- CTL
- Fair CTL

CTL^* (Emerson & Halpern 86)

Definition: Syntax of the Computation Tree Logic CTL^*

$$\varphi ::= \bot \mid p\ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \mathsf{E}\,\varphi \mid \mathsf{A}\,\varphi$$

In this chapter, temporal modalities U, F, G, ... are non-strict.
We may also add past modalities Y and S

Definition: Semantics of CTL^*

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure.
Let $\sigma = s_0 s_1 s_2 \cdots$ be an infinte run of M.

$M, \sigma, i \models \mathsf{E}\varphi$ if $M, \sigma', i \models \varphi$ for some infinite run σ' such that $\sigma'[i] = \sigma[i]$

$M, \sigma, i \models \mathsf{A}\varphi$ if $M, \sigma', i \models \varphi$ for all infinite runs σ' such that $\sigma'[i] = \sigma[i]$

where $\sigma[i] = s_0 \cdots s_i$.

Remark:

- $\mathsf{A}\,\varphi \equiv \neg\,\mathsf{E}\,\neg\varphi$
- $\sigma'[i] = \sigma[i]$ means that future is branching but past is not.

CTL^* (Emerson & Halpern 86)

Example: Some specifications

- EF φ: φ is possible
- AG φ: φ is an invariant
- AF φ: φ is unavoidable
- EG φ: φ holds globally along some path

State formulae and path formulae

Definition: State formulae

$\varphi \in \mathrm{CTL}^*$ is a state formula if $\forall M, \sigma, \sigma', i, j$ such that $\sigma(i) = \sigma'(j)$ we have

$$M, \sigma, i \models \varphi \iff M, \sigma', j \models \varphi$$

If φ is a state formula and $M = (S, T, I, \mathrm{AP}, \ell)$, define

$$[\![\varphi]\!]^M = \{s \in S \mid M, s \models \varphi\}$$

Example: State formulae

Atomic propositions are state formulae: $[\![p]\!] = \{s \in S \mid p \in \ell(s)\}$
State formulae are closed under boolean connectives.

$$[\![\neg\varphi]\!] = S \setminus [\![\varphi]\!] \qquad [\![\varphi_1 \vee \varphi_2]\!] = [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$$

Formulae of the form E φ or A φ are state formulae, provided φ is future.

Definition: Alternative syntax

State formulae $\varphi ::= \bot \mid p\ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}\,\psi \mid \mathsf{A}\,\psi$
Path formulae $\psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \mathsf{X}\,\psi \mid \psi\,\mathsf{U}\,\psi$

Model checking of CTL^*

Definition: Existential and universal model checking

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure and $\varphi \in \mathrm{CTL}^*$ a formula.

$M \models_\exists \varphi$ if $M, \sigma, 0 \models \varphi$ for some initial infinite run σ of M.
$M \models_\forall \varphi$ if $M, \sigma, 0 \models \varphi$ for all initial infinite run σ of M.

Remark:

$M \models_\exists \varphi$ iff $I \cap [\![\mathsf{E}\,\varphi]\!] \neq \emptyset$

$M \models_\forall \varphi$ iff $I \subseteq [\![\mathsf{A}\,\varphi]\!]$

$M \models_\forall \varphi$ iff $M \not\models_\exists \neg\varphi$

Definition: Model checking problems $\mathrm{MC}^\forall_{\mathrm{CTL}^*}$ and $\mathrm{MC}^\exists_{\mathrm{CTL}^*}$

Input: A Kripke structure $M = (S, T, I, \mathrm{AP}, \ell)$ and a formula $\varphi \in \mathrm{CTL}^*$

Question: Does $M \models_\forall \varphi$? or Does $M \models_\exists \varphi$?

Complexity of CTL^*

Definition: Syntax of the Computation Tree Logic CTL^*

$$\varphi ::= \bot \mid p\ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{X}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \mathsf{E}\,\varphi \mid \mathsf{A}\,\varphi$$

Theorem

The model checking problem for CTL^* is PSPACE-complete

Proof:

PSPACE-hardness: follows from $\mathrm{LTL} \subseteq \mathrm{CTL}^*$.

PSPACE-easiness: reduction to LTL-model checking by inductive eliminations of path quantifications.

$\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}$ in PSPACE

Proof:

For $\psi \in \mathrm{LTL}$, let $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M,t,\psi)$ be the function which computes in polynomial space whether $M,t \models_{\exists} \psi$, i.e., if $M,t \models \mathsf{E}\,\psi$.

Let $M = (S,T,I,\mathrm{AP},\ell)$ be a Kripke structure, $s \in S$ and $\varphi \in \mathrm{CTL}^*$.
Replacing $\mathsf{A}\,\psi$ by $\neg\,\mathsf{E}\,\neg\psi$ we assume φ only contains the existential path quantifier.

$\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}(M,s,\varphi)$

- If E does not occur in φ then return $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M,s,\varphi)$ fi
- Let $\mathsf{E}\,\psi$ be a subformula of φ with $\psi \in \mathrm{LTL}$
- Let e_ψ be a new propositional variable
- Define $\ell' : S \to 2^{\mathrm{AP}'}$ with $\mathrm{AP}' = \mathrm{AP} \uplus \{e_\psi\}$ by
  - $\ell'(t) \cap \mathrm{AP} = \ell(t)$ and $e_\psi \in \ell'(t)$ iff $\mathrm{MC}^{\exists}_{\mathrm{LTL}}(M,t,\psi)$
- Let $M' = (S,T,I,\mathrm{AP}',\ell')$
- Let $\varphi' = \varphi[e_\psi / \mathsf{E}\,\psi]$ be obtained from φ by replacing each $\mathsf{E}\,\psi$ by e_ψ
- Return $\mathrm{MC}^{\exists}_{\mathrm{CTL}^*}(M',s,\varphi')$

Satisfiability for CTL^*

Definition: $\mathrm{SAT}(\mathrm{CTL}^*)$

Input: A formula $\varphi \in \mathrm{CTL}^*$

Question: Existence of a model M and a run σ such that $M,\sigma,0 \models \varphi$?

Theorem

The satisfiability problem for CTL^* is 2-EXPTIME-complete

Outline

CTL (Clarke & Emerson 81)

Definition: Computation Tree Logic (CTL)

Syntax:

$$\varphi ::= \bot \mid p\ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{EX}\,\varphi \mid \mathsf{AX}\,\varphi \mid \mathsf{E}\,\varphi\,\mathsf{U}\,\varphi \mid \mathsf{A}\,\varphi\,\mathsf{U}\,\varphi$$

The semantics is inherited from CTL^*.

Remark: All CTL formulae are state formulae

$$[\![\varphi]\!]^M = \{s \in S \mid M,s \models \varphi\}$$

Examples: Macros

- $\mathsf{EF}\,\varphi = \mathsf{E}\,\top\,\mathsf{U}\,\varphi$ and $\mathsf{AF}\,\varphi = \mathsf{A}\,\top\,\mathsf{U}\,\varphi$
- $\mathsf{EG}\,\varphi = \neg\,\mathsf{AF}\,\neg\varphi$ and $\mathsf{AG}\,\varphi = \neg\,\mathsf{EF}\,\neg\varphi$
- $\mathsf{AG}(\mathrm{req} \to \mathsf{EF}\,\mathrm{grant})$
- $\mathsf{AG}(\mathrm{req} \to \mathsf{AF}\,\mathrm{grant})$

CTL (Clarke & Emerson 81)

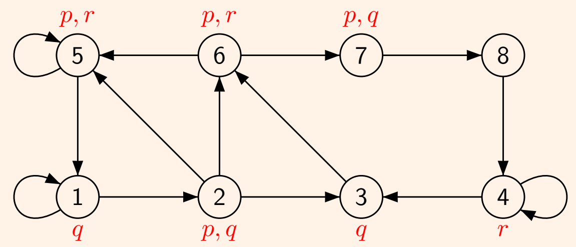
Definition: Semantics

All CTL-formulae are state formulae. Hence, we have a simpler semantics.
Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure without deadlocks and let $s \in S$.

$s \models p$	if	$p \in \ell(s)$
$s \models \mathsf{EX}\,\varphi$	if	$\exists s \to s'$ with $s' \models \varphi$
$s \models \mathsf{AX}\,\varphi$	if	$\forall s \to s'$ we have $s' \models \varphi$
$s \models \mathsf{E}\,\varphi\,\mathsf{U}\,\psi$	if	$\exists s = s_0 \to s_1 \to s_2 \to \cdots s_j$ finite path, with $s_j \models \psi$ and $s_k \models \varphi$ for all $0 \le k < j$
$s \models \mathsf{A}\,\varphi\,\mathsf{U}\,\psi$	if	$\forall s = s_0 \to s_1 \to s_2 \to \cdots$ infinite path, $\exists j \geq 0$ with $s_j \models \psi$ and $s_k \models \varphi$ for all $0 \le k < j$

CTL (Clarke & Emerson 81)

Example:



$[\![\mathsf{EX}\,p]\!] =$

$[\![\mathsf{AX}\,p]\!] =$

$[\![\mathsf{EF}\,p]\!] =$

$[\![\mathsf{AF}\,p]\!] =$

$[\![\mathsf{E}\,q\,\mathsf{U}\,r]\!] =$

$[\![\mathsf{A}\,q\,\mathsf{U}\,r]\!] =$

CTL (Clarke & Emerson 81)

Remark: Equivalent formulae

- $\mathsf{AX}\,\varphi = \neg\,\mathsf{EX}\,\neg\varphi$,
- $\neg(\varphi\,\mathsf{U}\,\psi) = \mathsf{G}\,\neg\psi \vee (\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$
- $\mathsf{A}\,\varphi\,\mathsf{U}\,\psi = \neg\,\mathsf{EG}\,\neg\psi \wedge \neg\,\mathsf{E}(\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$
- $\mathsf{AG}(\mathrm{req} \to \mathsf{F}\,\mathrm{grant}) = \mathsf{AG}(\mathrm{req} \to \mathsf{AF}\,\mathrm{grant})$
- $\mathsf{A}\,\mathsf{G}\,\mathsf{F}\,\varphi = \mathsf{AG}\,\mathsf{AF}\,\varphi$ — infinitely often
- $\mathsf{E}\,\mathsf{F}\,\mathsf{G}\,\varphi = \mathsf{EF}\,\mathsf{EG}\,\varphi$ — ultimately
- $\mathsf{EG}\,\mathsf{EF}\,\varphi \neq \mathsf{E}\,\mathsf{G}\,\mathsf{F}\,\varphi$
- $\mathsf{AF}\,\mathsf{AG}\,\varphi \neq \mathsf{A}\,\mathsf{F}\,\mathsf{G}\,\varphi$
- $\mathsf{EG}\,\mathsf{EX}\,\varphi \neq \mathsf{E}\,\mathsf{G}\,\mathsf{X}\,\varphi$

Model checking of CTL

Definition: Existential and universal model checking

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure and $\varphi \in \mathrm{CTL}$ a formula.

$M \models_\exists \varphi$ if $M, s \models \varphi$ for some $s \in I$.
$M \models_\forall \varphi$ if $M, s \models \varphi$ for all $s \in I$.

Remark:

$M \models_\exists \varphi$ iff $I \cap [\![\varphi]\!] \neq \emptyset$

$M \models_\forall \varphi$ iff $I \subseteq [\![\varphi]\!]$

$M \models_\forall \varphi$ iff $M \not\models_\exists \neg\varphi$

Definition: Model checking problems $\mathrm{MC}^\forall_{\mathrm{CTL}}$ and $\mathrm{MC}^\exists_{\mathrm{CTL}}$

Input: A Kripke structure $M = (S, T, I, \mathrm{AP}, \ell)$ and a formula $\varphi \in \mathrm{CTL}$

Question: Does $M \models_\forall \varphi$? or Does $M \models_\exists \varphi$?

Model checking of CTL

Theorem

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure and $\varphi \in \mathrm{CTL}$ a formula.
The model checking problem $M \models_\exists \varphi$ is decidable in time $\mathcal{O}(|M| \cdot |\varphi|)$

Proof:

Compute $[\![\varphi]\!] = \{s \in S \mid M, s \models \varphi\}$ by induction on the formula.

The set $[\![\varphi]\!]$ is represented by a boolean array: $L[s][\varphi] = \top$ if $s \in [\![\varphi]\!]$.

The labelling ℓ is encoded in L: for $p \in \mathrm{AP}$ we have $L[s][p] = \top$ if $p \in \ell(s)$.

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = \neg\varphi_1$
  semantics(φ_1)
  $[\![\varphi]\!] := S \setminus [\![\varphi_1]\!]$ — $\mathcal{O}(|S|)$

case $\varphi = \varphi_1 \vee \varphi_2$
  semantics(φ_1); semantics(φ_2)
  $[\![\varphi]\!] := [\![\varphi_1]\!] \cup [\![\varphi_2]\!]$ — $\mathcal{O}(|S|)$

case $\varphi = EX\varphi_1$
  semantics(φ_1)
  $[\![\varphi]\!] := \emptyset$ — $\mathcal{O}(|S|)$
  for all $(s,t) \in T$ do if $t \in [\![\varphi_1]\!]$ then $[\![\varphi]\!] := [\![\varphi]\!] \cup \{s\}$ — $\mathcal{O}(|T|)$

case $\varphi = AX\varphi_1$
  semantics(φ_1)
  $[\![\varphi]\!] := S$ — $\mathcal{O}(|S|)$
  for all $(s,t) \in T$ do if $t \notin [\![\varphi_1]\!]$ then $[\![\varphi]\!] := [\![\varphi]\!] \setminus \{s\}$ — $\mathcal{O}(|T|)$

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = E\varphi_1 \mathsf{U} \varphi_2$ — $\mathcal{O}(|S| + |T|)$
  semantics(φ_1); semantics(φ_2)
  $L := [\![\varphi_2]\!]$ // the "todo" set L is imlemented with a list — $\mathcal{O}(|S|)$
  $Z := [\![\varphi_2]\!]$ // the "result" is computed in the array Z — $\mathcal{O}(|S|)$
  while $L \neq \emptyset$ do — $|S|$ times
  Invariant: $L \subseteq Z$ and
    $[\![\varphi_2]\!] \cup ([\![\varphi_1]\!] \cap T^{-1}(Z \setminus L)) \subseteq Z \subseteq [\![\mathsf{E}\,\varphi_1 \mathsf{U} \varphi_2]\!]$
    take $t \in L$; $L := L \setminus \{t\}$ — $\mathcal{O}(1)$
    for all $s \in T^{-1}(t)$ do — $|T|$ times
      if $s \in [\![\varphi_1]\!] \setminus Z$ then $L := L \cup \{s\}$; $Z := Z \cup \{s\}$ — $\mathcal{O}(1)$
  od
  $[\![\varphi]\!] := Z$ — $\mathcal{O}(|S|)$

Z is only used to make the invariant clear. It can be replaced by $[\![\varphi]\!]$.

Model checking of CTL

Definition: procedure semantics(φ)

case $\varphi = A\varphi_1 \mathsf{U} \varphi_2$ — $\mathcal{O}(|S| + |T|)$
  semantics(φ_1); semantics(φ_2)
  $L := [\![\varphi_2]\!]$ // the "todo" set L is imlemented with a list — $\mathcal{O}(|S|)$
  $Z := [\![\varphi_2]\!]$ // the "result" is computed in the array Z — $\mathcal{O}(|S|)$
  for all $s \in S$ do $c[s] := |T(s)|$ — $\mathcal{O}(|S|)$
  while $L \neq \emptyset$ do — $|S|$ times
  Invariant: $L \subseteq Z$ and
    $\forall s \in S$, $c[s] = |T(s) \setminus (Z \setminus L)|$ and
    $[\![\varphi_2]\!] \cup ([\![\varphi_1]\!] \cap \{s \in S \mid c[s] = 0\}) \subseteq Z \subseteq [\![\mathsf{A}\,\varphi_1 \mathsf{U} \varphi_2]\!]$
    take $t \in L$; $L := L \setminus \{t\}$ — $\mathcal{O}(1)$
    for all $s \in T^{-1}(t)$ do — $|T|$ times
      $c[s] := c[s] - 1$ — $\mathcal{O}(1)$
      if $c[s] = 0 \wedge s \in [\![\varphi_1]\!] \setminus Z$ then $L := L \cup \{s\}$; $Z := Z \cup \{s\}$ — $\mathcal{O}(1)$
  od
  $[\![\varphi]\!] := Z$ — $\mathcal{O}(|S|)$

Z is only used to make the invariant clear. It can be replaced by $[\![\varphi]\!]$.

Complexity of CTL

Definition: $\mathrm{SAT}(\mathrm{CTL})$

Input: A formula $\varphi \in \mathrm{CTL}$

Question: Existence of a model M and a state s such that $M, s \models \varphi$?

Theorem: Complexity

- The model checking problem for CTL is PTIME-complete.
- The satisfiability problem for CTL is EXPTIME-complete.

Outline

fairness

Example: Fairness

Only fair runs are of interest

- Each process is enabled infinitely often: $\bigwedge_i \mathsf{G}\,\mathsf{F}\,\mathrm{run}_i$
- No process stays ultimately in the critical section: $\bigwedge_i \neg\,\mathsf{F}\,\mathsf{G}\,\mathrm{CS}_i = \bigwedge_i \mathsf{G}\,\mathsf{F}\,\neg\mathrm{CS}_i$

Definition: Fair Kripke structure

$M = (S, T, I, \mathrm{AP}, \ell, F_1, \ldots, F_n)$ with $F_i \subseteq S$.

An infinite run σ is fair if it visits infinitely often each F_i

fair CTL

Definition: Syntax of fair-CTL

$\varphi ::= \bot \mid p\ (p \in \mathrm{AP}) \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}_f\,\mathsf{X}\,\varphi \mid \mathsf{A}_f\,\mathsf{X}\,\varphi \mid \mathsf{E}_f\,\varphi\,\mathsf{U}\,\varphi \mid \mathsf{A}_f\,\varphi\,\mathsf{U}\,\varphi$

Definition: Semantics as a fragment of CTL^*

Let $M = (S, T, I, \mathrm{AP}, \ell, F_1, \ldots, F_n)$ be a fair Kripke structure.

Then, $\mathsf{E}_f\,\varphi = \mathsf{E}(\mathrm{fair} \wedge \varphi)$ and $\mathsf{A}_f\,\varphi = \mathsf{A}(\mathrm{fair} \to \varphi)$

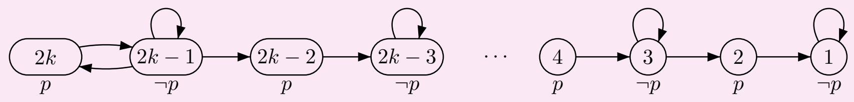
where $\mathrm{fair} = \bigwedge_i \mathsf{G}\,\mathsf{F}\,F_i$

Lemma: CTL_f cannot be expressed in CTL

fair CTL

Proof: CTL_f cannot be expressed in CTL

Consider the Kripke structure M_k defined by:



- $M_k, 2k \models \mathsf{E\,G\,F}\,p$ but $M_k, 2k-2 \not\models \mathsf{E\,G\,F}\,p$
- If $\varphi \in \mathrm{CTL}$ and $|\varphi| \leq m \leq k$ then

$$M_k, 2k \models \varphi \text{ iff } M_k, 2m \models \varphi$$

$$M_k, 2k-1 \models \varphi \text{ iff } M_k, 2m-1 \models \varphi$$

If the fairness condition is $\ell^{-1}(p)$ then $\mathsf{E}_f \top$ cannot be expressed in CTL.

Model checking of CTL_f

Theorem

The model checking problem for CTL_f is decidable in time $\mathcal{O}(|M| \cdot |\varphi|)$

Proof: Computation of $\mathrm{Fair} = \{s \in S \mid M, s \models \mathsf{E}_f \top\}$

Compute the SCC of M with Tarjan's algorithm (in time $\mathcal{O}(|M|)$).
Let S' be the union of the (non trivial) SCCs which intersect each F_i.
Then, Fair is the set of states that can reach S'.
Note that reachability can be computed in linear time.

Model checking of CTL_f

Proof: Reductions

$\mathsf{E}_f\,\mathsf{X}\,\varphi = \mathsf{E\,X}(\mathrm{Fair} \wedge \varphi)$ and $\mathsf{E}_f\,\varphi\,\mathsf{U}\,\psi = \mathsf{E}\,\varphi\,\mathsf{U}\,(\mathrm{Fair} \wedge \psi)$
It remains to deal with $\mathsf{A}_f\,\varphi\,\mathsf{U}\,\psi$.
We have $\mathsf{A}_f\,\varphi\,\mathsf{U}\,\psi = \neg\,\mathsf{E}_f\,\mathsf{G}\,\neg\psi \wedge \neg\,\mathsf{E}_f(\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$
Hence, we only need to compute the semantics of $\mathsf{E}_f\,\mathsf{G}\,\varphi$.

Proof: Computation of $\mathsf{E}_f\,\mathsf{G}\,\varphi$

Let M_φ be the restriction of M to $[\![\varphi]\!]_f$.
Compute the SCC of M_φ with Tarjan's algorithm (in linear time).
Let S' be the union of the (non trivial) SCCs of M_φ which intersect each F_i.
Then, $M, s \models \mathsf{E}_f\,\mathsf{G}\,\varphi$ iff $M, s \models \mathsf{E}\,\varphi\,\mathsf{U}\,S'$ iff $M_\varphi, s \models \mathsf{EF}\,S'$.
This is again a reachability problem which can be solved in linear time.