IA169 Model Checking

CTL model checking

Jan Strejček

Faculty of Informatics
Masaryk University

Linear vs. branching time

linear time view

- Amir Pnueli, 1977
- system behavior can be seen as a set of state sequences
- property is a restriction applied to each such a sequence
- property can be described by LTL

Linear vs. branching time

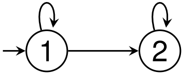
linear time view

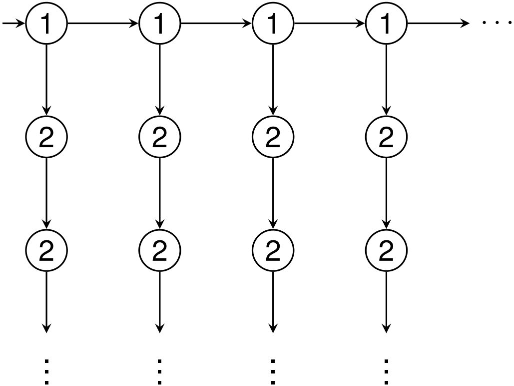
- Amir Pnueli, 1977
- system behavior can be seen as a set of state sequences
- property is a restriction applied to each such a sequence
- property can be described by LTL

branching time view

- Edmund M. Clarke and E. Allen Emerson, 1980
- system behavior is a computation tree, i.e., a branching structure of possible successors of each reachable state of the system
- property is a restriction on the tree
- property can be described by CTL or CTL*

Example of a system and its computation tree





Agenda and sources

agenda

- computation tree logic (CTL)
- CTL model checking
- CTL*

source

- Chapters 5 and 6 of *E. M. Clarke, O. Grumberg, D. Kroening, D. Peled, and R. Bloem: Model Checking, Second Edition, MIT, 2018.*

Computation tree logic (CTL)

Intuition for CTL

- we present only state-based CTL model checking
- for a given in node of a computation tree, the subtree rooted by the node represents all possible runs from the node
- CTL formula talks about runs from this node
- CTL uses temporal operators X, U, F, G known from LTL, but extended with quantifier A saying that the formula should hold on all runs or quantifier E saying that there exists a run satisfying the formula
- for example, EFa says that there exists a run from the node such that a holds somewhere on the run

Computation tree logic (CTL)

Definition (computation tree logic, CTL)

Formulae of Computation Tree Logic (CTL) are defined by

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{EX}\varphi \mid \mathsf{E}(\varphi_1 \, \mathsf{U} \, \varphi_2) \mid \mathsf{A}(\varphi_1 \, \mathsf{U} \, \varphi_2)$$

where $\top$ stands for true and a ranges over a countable set AP.
By $AP(\varphi)$ we denote the set of atomic propositions appearing in φ.

Computation tree logic (CTL)

Definition (computation tree logic, CTL)

Formulae of Computation Tree Logic (CTL) are defined by

$$\varphi \ ::= \ \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{EX}\varphi \mid \mathsf{E}(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \mathsf{A}(\varphi_1 \,\mathsf{U}\, \varphi_2)$$

where $\top$ stands for true and a ranges over a countable set AP.
By $AP(\varphi)$ we denote the set of atomic propositions appearing in φ.

abbreviations

- standard ones for $\bot, \vee, \Rightarrow, \Leftrightarrow$
- $\mathsf{EF}\varphi \equiv \mathsf{E}(\top \,\mathsf{U}\, \varphi)$
- $\mathsf{AF}\varphi \equiv \mathsf{A}(\top \,\mathsf{U}\, \varphi)$
- $\mathsf{EG}\varphi \equiv \neg\mathsf{AF}\neg\varphi$
- $\mathsf{AG}\varphi \equiv \neg\mathsf{EF}\neg\varphi$
- $\mathsf{AX}\varphi \equiv \neg\mathsf{EX}\neg\varphi$

Intuitive semantic of CTL

$\mathrm{EX}a$

$\mathrm{AX}a$

$\mathrm{E}(a \,\mathrm{U}\, b)$

$\mathrm{A}(a \,\mathrm{U}\, b)$

$\mathrm{EF}a$

$\mathrm{AF}a$

$\mathrm{EG}a$

$\mathrm{AG}a$

Semantics of CTL

- we interpret CTL over states of a Kripke structure
- we assume that each state of a Kripke structure has at least one successor

Definition (path)

Let $K = (S, T, S_0, L)$ be a Kripke structure and $s \in S$ be its state. An (infinite) path of K starting in s is an infinite sequence $\pi = s_0 s_1 s_2 \ldots$ of states such that $s_0 = s$ and $(s_i, s_{i+1}) \in T$ holds for each $i \geq 0$.

By $\pi(i)$ we denote the state s_i of π.

By π_i we denote the infinite path $\pi(i)\pi(i+1)\pi(i+2)\ldots$.

By $P_K(s)$ we denote the set of all infinite paths starting in s.

Semantics of CTL

Definition

The relation $K, s \models \varphi$, meaning that state s of a Kripke structure $K = (S, T, S_0, L)$ satisfies CTL formula φ, is defined inductively as follows.

$$
\begin{array}{lll}
K, s \models \top & & \\
K, s \models a & \text{iff} & a \in L(s) \\
K, s \models \neg\varphi & \text{iff} & K, s \not\models \varphi \\
K, s \models \varphi_1 \wedge \varphi_2 & \text{iff} & K, s \models \varphi_1 \wedge K, s \models \varphi_2 \\
K, s \models \mathrm{EX}\varphi & \text{iff} & \exists \pi \in P_K(s) \,.\, K, \pi(1) \models \varphi \\
K, s \models \mathrm{E}(\varphi_1 \,\mathrm{U}\, \varphi_2) & \text{iff} & \exists \pi \in P_K(s) \,.\, \exists i \geq 0 \,.\, K, \pi(i) \models \varphi_2 \wedge \\
 & & \qquad \wedge \forall 0 \leq j < i \,.\, K, \pi(j) \models \varphi_1 \\
K, s \models \mathrm{A}(\varphi_1 \,\mathrm{U}\, \varphi_2) & \text{iff} & \forall \pi \in P_K(s) \,.\, \exists i \geq 0 \,.\, K, \pi(i) \models \varphi_2 \wedge \\
 & & \qquad \wedge \forall 0 \leq j < i \,.\, K, \pi(j) \models \varphi_1
\end{array}
$$

Semantics of CTL

Definition

The relation $K, s \models \varphi$, meaning that state s of a Kripke structure $K = (S, T, S_0, L)$ satisfies CTL formula φ, is defined inductively as follows.

$$
\begin{array}{lll}
K, s \models \top & & \\
K, s \models a & \text{iff} & a \in L(s) \\
K, s \models \neg\varphi & \text{iff} & K, s \not\models \varphi \\
K, s \models \varphi_1 \wedge \varphi_2 & \text{iff} & K, s \models \varphi_1 \wedge K, s \models \varphi_2 \\
K, s \models \mathrm{EX}\varphi & \text{iff} & \exists \pi \in P_K(s) \,.\, K, \pi(1) \models \varphi \\
K, s \models \mathrm{E}(\varphi_1 \,\mathrm{U}\, \varphi_2) & \text{iff} & \exists \pi \in P_K(s) \,.\, \exists i \geq 0 \,.\, K, \pi(i) \models \varphi_2 \wedge \\
& & \qquad \wedge \forall 0 \leq j < i \,.\, K, \pi(j) \models \varphi_1 \\
K, s \models \mathrm{A}(\varphi_1 \,\mathrm{U}\, \varphi_2) & \text{iff} & \forall \pi \in P_K(s) \,.\, \exists i \geq 0 \,.\, K, \pi(i) \models \varphi_2 \wedge \\
& & \qquad \wedge \forall 0 \leq j < i \,.\, K, \pi(j) \models \varphi_1
\end{array}
$$

K satisfies φ, written $K \models \varphi$, if $K, s_0 \models \varphi$ holds for every $s_0 \in S_0$.

Exercise

- condsider a Kripke structure with atomic propositions $\{a, b, r, restart\}$
- express the following properties by CTL formulae
  1. it is possible to reach a state where a holds and b does not
  2. whenever request r is received, the system eventually generates acknowledgment a
  3. whenever b holds, it is possible that b will never hold again
  4. there is always an option to reset by system, i.e., to reach a state where *restart* holds

CTL model checking

CTL model checking problems

Let $K = (S, T, S_0, L)$ be a Kripke structure and φ be a CTL formula. We can consider the following problems.

- to decide whether $K \models \varphi$
- local CTL model checking problem: to decide whether $K, s \models \varphi$ holds for a given state $s \in S$
- global CTL model checking problem: to compute the set of states where φ holds, i.e., the set $\{s \in S \mid K, s \models \varphi\}$.

CTL model checking problems

Let $K = (S, T, S_0, L)$ be a Kripke structure and φ be a CTL formula. We can consider the following problems.

- to decide whether $K \models \varphi$
- local CTL model checking problem: to decide whether $K, s \models \varphi$ holds for a given state $s \in S$
- global CTL model checking problem: to compute the set of states where φ holds, i.e., the set $\{s \in S \mid K, s \models \varphi\}$.

We present an algorithm that can decide all the problems on finite Kripke structures. Since now on, we consider only Kripke structures with finitely many states.

Idea of the algorithm

- let $K = (S, T, S_0, L)$ be a Kripke structure and φ be a CTL formula
- we transform φ to the form that uses only existentially quantified temporal operators EX, EG, EU (i.e., not AU) using the equivalence

$$\mathrm{A}(\varphi\,\mathrm{U}\,\psi) \;\equiv\; \neg\mathrm{EG}\neg\psi \;\wedge\; \neg\mathrm{E}(\neg\psi\,\mathrm{U}\,(\neg\varphi\wedge\neg\psi))$$

- hence, we assume that φ is of the form

$$\varphi \;::=\; \top \mid a \mid \neg\varphi \mid \varphi_1\wedge\varphi_2 \mid \mathrm{EX}\varphi \mid \mathrm{EG}\varphi \mid \mathrm{E}(\varphi_1\,\mathrm{U}\,\varphi_2)$$

- let $\mathit{subf}(\varphi)$ denote all subformulae of φ, for example $\mathit{subf}(\mathrm{E}(\neg a\,\mathrm{U}\,\mathrm{EG}(b\wedge c))) = \{\mathrm{E}(\neg a\,\mathrm{U}\,\mathrm{EG}(b\wedge c)), \neg a, a, \mathrm{EG}(b\wedge c), b\wedge c, b, c\}$
- the algorithm computes function label $: S \to 2^{\mathit{subf}(\varphi)}$ assigning to each state s the set of all subformulae ψ satisfying $K, s \models \psi$
- the function is built gradually, starting with the atomic proposition of φ and proceeding towards more complex subformulae, ending with φ itself

CTL model checking algorithm

input : a Kripke structure $K = (S, T, S_0, L)$ and a CTL formula φ
output: function label : $S \to 2^{subf(\varphi)}$ satisfying $\varphi \in \text{label}(s)$ iff $K, s \models \varphi$ for each $s \in S$

procedure CTLmc(K, φ)
  forall $s \in S$ **do** label(s) $\leftarrow (L(s) \cap AP(\varphi)) \cup (\{\top\} \cap subf(\varphi))$
  $solved \leftarrow AP(\varphi) \cup (\{\top, \bot\} \cap subf(\varphi))$
  while $\varphi \notin$ solved **do**
    choose $\psi \in subf(\varphi) \smallsetminus$ solved such that $subf(\psi) \smallsetminus \{\psi\} \subseteq$ solved
    updateLabel(ψ)
    solved $\leftarrow$ solved $\cup \{\psi\}$
  return label

procedure updateLabel(ψ)
  if $\psi \equiv \mathrm{E}(\rho_1 \mathbin{\mathrm{U}} \rho_2)$ **then** checkEU(ρ_1, ρ_2)
  if $\psi \equiv \mathrm{EG}\rho$ **then** checkEG(ρ)
  forall $s \in S$ **do**
    if $\psi \equiv \neg\rho$ and $\rho \notin$ label(s) **then** label(s) $\leftarrow$ label(s) $\cup \{\psi\}$
    if $\psi \equiv \rho_1 \wedge \rho_2$ and $\rho_1, \rho_2 \in$ label(s) **then** label(s) $\leftarrow$ label(s) $\cup \{\psi\}$
    if $\psi \equiv \mathrm{EX}\rho$ and there exists $s' \in S$ such that $(s, s') \in T$ and $\rho \in$ label(s') **then**
      label(s) $\leftarrow$ label(s) $\cup \{\psi\}$

CTL model checking algorithm

procedure checkEU(ρ_1, ρ_2)
- $Q \leftarrow \{s \mid \rho_2 \in \text{label}(s)\}$
- **forall** $s \in Q$ **do** $\text{label}(s) \leftarrow \text{label}(s) \cup \{\text{E}(\rho_1 \,\text{U}\, \rho_2)\}$
- **while** $Q \neq \emptyset$ **do**
  - choose $s \in Q$
  - $Q \leftarrow Q \setminus \{s\}$
  - **forall** s' such that $(s', s) \in T$ **do**
    - **if** $\rho_1 \in \text{label}(s')$ and $\text{E}(\rho_1 \,\text{U}\, \rho_2) \notin \text{label}(s')$ **then**
      - $\text{label}(s') \leftarrow \text{label}(s') \cup \{\text{E}(\rho_1 \,\text{U}\, \rho_2)\}$
      - $Q \leftarrow Q \cup \{s'\}$

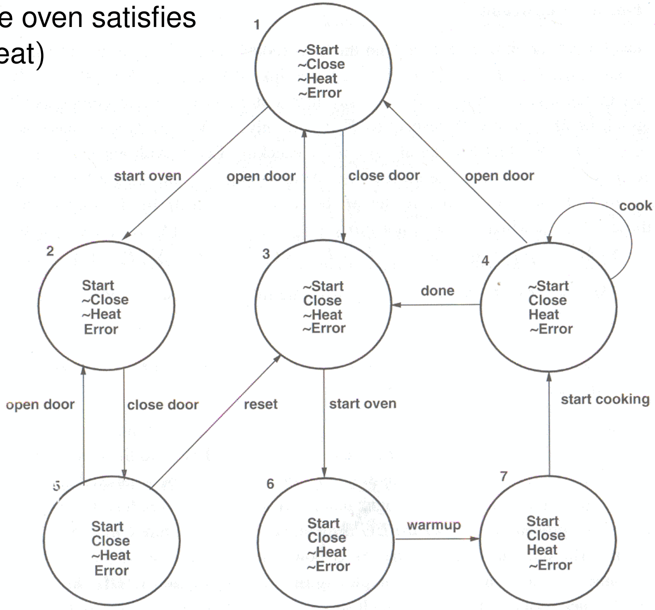
CTL model checking algorithm

procedure checkEG(ρ)

$S' \leftarrow \{s \mid \rho \in \text{label}(s)\}$
$Q \leftarrow \{s \mid s \text{ is a node of some nontrivial SCC of graph } (S', T \cap (S' \times S'))\}$
forall $s \in Q$ **do** $\text{label}(s) \leftarrow \text{label}(s) \cup \{\text{EG}\rho\}$
while $Q \neq \emptyset$ **do**
    choose $s \in Q$
    $Q \leftarrow Q \setminus \{s\}$
    forall s' such that $(s', s) \in T$ **do**
        if $\rho \in \text{label}(s')$ and $\text{EG}\rho \notin \text{label}(s')$ **then**
            $\text{label}(s') \leftarrow \text{label}(s') \cup \{\text{EG}\rho\}$
            $Q \leftarrow Q \cup \{s'\}$

Example

check if microwave oven satisfies
$AG(Start \Rightarrow AF\,Heat)$



Transformation of formula $\mathrm{AG}(\mathrm{Start} \Rightarrow \mathrm{AF\,Heat})$

$$
\begin{aligned}
\mathrm{AG}(\mathrm{Start} \Rightarrow \mathrm{AF\,Heat}) &\equiv \neg\mathrm{EF}(\neg(\mathrm{Start} \Rightarrow \mathrm{AF\,Heat})) \\
&\equiv \neg\mathrm{EF}(\mathrm{Start} \wedge \neg\mathrm{AF\,Heat}) \\
&\equiv \neg\mathrm{EF}(\mathrm{Start} \wedge \mathrm{EG}\neg\mathrm{Heat}) \\
&\equiv \neg\mathrm{E}(\top\,\mathrm{U}\,(\mathrm{Start} \wedge \mathrm{EG}\neg\mathrm{Heat}))
\end{aligned}
$$

Transformation of formula AG(Start $\Rightarrow$ AF Heat)

$$
\begin{aligned}
\mathrm{AG}(\mathrm{Start} \Rightarrow \mathrm{AF\,Heat}) &\equiv \neg \mathrm{EF}(\neg(\mathrm{Start} \Rightarrow \mathrm{AF\,Heat})) \\
&\equiv \neg \mathrm{EF}(\mathrm{Start} \wedge \neg \mathrm{AF\,Heat}) \\
&\equiv \neg \mathrm{EF}(\mathrm{Start} \wedge \mathrm{EG} \neg \mathrm{Heat}) \\
&\equiv \neg \mathrm{E}(\top\, \mathrm{U}\, (\mathrm{Start} \wedge \mathrm{EG} \neg \mathrm{Heat}))
\end{aligned}
$$

subformuala ρ	states satisfying ρ, i.e. $\{s \mid K, s \models \rho\}$
$\top$	$\{1,2,3,4,5,6,7\}$
Start	$\{2,5,6,7\}$
Heat	$\{4,7\}$
$\neg$Heat	$\{1,2,3,5,6\}$
EG$\neg$Heat	$\{1,2,3,5\}$
Start $\wedge$ EG$\neg$Heat	$\{2,5\}$
$\top$ U (Start $\wedge$ EG$\neg$Heat)	$\{1,2,3,4,5,6,7\}$
$\neg$E($\top$ U (Start $\wedge$ EG$\neg$Heat))	$\emptyset$

Complexity of the CTL model checking algorithm

- each formula φ has at most $|\varphi|$ subformulae
- decomposition of every subgraph $(S', T \cap (S' \times S'))$ of K into SCCs can be done in time $\mathcal{O}(|S| + |T|)$
- every call of updateLabel(ψ) terminates in time $\mathcal{O}(|S| + |T|)$
- CTLmc runs in time $\mathcal{O}(|\varphi| \cdot (|S| + |T|))$ and in space $\mathcal{O}(|\varphi| \cdot |S|)$

Complexity of the CTL model checking algorithm

- each formula φ has at most $|\varphi|$ subformulae
- decomposition of every subgraph $(S', T \cap (S' \times S'))$ of K into SCCs can be done in time $\mathcal{O}(|S| + |T|)$
- every call of updateLabel(ψ) terminates in time $\mathcal{O}(|S| + |T|)$
- CTLmc runs in time $\mathcal{O}(|\varphi| \cdot (|S| + |T|))$ and in space $\mathcal{O}(|\varphi| \cdot |S|)$

- despite its linear complexity, the algorithm also suffers from state-space explosion as the Kripke structure can be extremely large
- in fact, the problem is common for all explicit-state model checking algorithms, where states are handled individually

CTL*

Comparison of LTL and CTL

- LTL and CTL are expressively incomparable
- there is no CTL formula φ such that $K \models \varphi \iff K \models \mathrm{FG}\, a$ for each K
- there is no LTL formula φ such that $K \models \varphi \iff K \models \mathrm{AGEF}\, a$ for each K

Comparison of LTL and CTL

- LTL and CTL are expressively incomparable
- there is no CTL formula φ such that $K \models \varphi \iff K \models \mathrm{FG}\, a$ for each K
- there is no LTL formula φ such that $K \models \varphi \iff K \models \mathrm{AGEF}\, a$ for each K

CTL*

- a common generalization of both CTL and LTL
- a branching time logic
- the main idea is to decouple temporal operators and quantifiers
- for example, $\mathrm{A}(a \wedge \mathrm{FG}\, b)$ is a CTL* formula, but not CTL formula

CTL*

- the syntax distinguishes two types of formulae: path and state formulae
- aplication of quantifiers E, A on a path formula results in a state formula

Definition (CTL*)

Formulae of CTL* are inductively defined by

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{E}\psi \qquad \text{(state formuale)}$$

$$\psi ::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \mathsf{X}\psi \mid \psi_1 \,\mathsf{U}\, \psi_2 \qquad \text{(path formulae)}$$

where $\top$ stands for true and a ranges over a countable set AP, φ represents state formulae and ψ represents path formulae.

CTL*

- the syntax distinguishes two types of formulae: path and state formulae
- aplication of quantifiers E, A on a path formula results in a state formula

Definition (CTL*)

Formulae of CTL* are inductively defined by

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{E}\psi \qquad \text{(state formuale)}$$

$$\psi ::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \mathsf{X}\psi \mid \psi_1 \,\mathsf{U}\, \psi_2 \qquad \text{(path formulae)}$$

where $\top$ stands for true and a ranges over a countable set AP, φ represents state formulae and ψ represents path formulae.

- similar abbreviations can be defined as for LTL and CTL

CTL*

- the syntax distinguishes two types of formulae: path and state formulae
- aplication of quantifiers E, A on a path formula results in a state formula

Definition (CTL*)

Formulae of CTL* are inductively defined by

$$\varphi \; ::= \; \top \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{E}\psi \qquad \text{(state formuale)}$$

$$\psi \; ::= \; \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \mathsf{X}\psi \mid \psi_1 \,\mathsf{U}\, \psi_2 \qquad \text{(path formulae)}$$

where $\top$ stands for true and a ranges over a countable set AP, φ represents state formulae and ψ represents path formulae.

- similar abbreviations can be defined as for LTL and CTL

- path/state formulae are interpreted over paths/states in a Kripke structure
- we assume that each state of a Kripke structure has at least one successor

Semantics of CTL*

- $\pi(i)$ denotes the $(i+1)$-st state of π and π_i denotes the path $\pi(i)\pi(i+1)\dots$

Definition

Let K be a Kripke structure, s be its state, and π be its infinite path. The relations $K, s \models \varphi$, meaning that state s satisfies a state formula φ, and $K, \pi \models \psi$, meaning that path π satisfies a path formula ψ, are defined inductively as follows.

$$
\begin{array}{lll}
K, s \models \top & & \\
K, s \models a & \text{iff} & a \in L(s) \\
K, s \models \neg\varphi & \text{iff} & K, s \not\models \varphi \\
K, s \models \varphi_1 \wedge \varphi_2 & \text{iff} & K, s \models \varphi_1 \wedge K, s \models \varphi_2 \\
K, s \models \mathsf{E}\psi & \text{iff} & \exists \pi \in P_K(s) \,.\, K, \pi \models \psi \\
K, \pi \models \varphi & \text{iff} & K, \pi(0) \models \varphi \\
K, \pi \models \neg\psi & \text{iff} & K, \pi \not\models \psi \\
K, \pi \models \psi_1 \wedge \psi_2 & \text{iff} & K, \pi \models \psi_1 \wedge K, \pi \models \psi_2 \\
K, \pi \models \mathsf{X}\psi & \text{iff} & K, \pi_1 \models \psi \\
K, \pi \models \psi_1 \,\mathsf{U}\, \psi_2 & \text{iff} & \exists i \geq 0 \,.\, K, \pi_i \models \psi_2 \;\wedge\; \forall 0 \leq j < i \,.\, K, \pi_j \models \psi_1
\end{array}
$$