Weaknesses of linear behaviors

Example:

$\varphi$: Whenever $p$ holds, it is possible to reach a state where $q$ holds.

$\varphi$ cannot be checked on linear behaviors.

We need to consider the computation-trees.

Remark: FO definable on the computation tree

$$\forall x \ (p(x) \rightarrow \exists y \ (x < y \land q(y)))$$
Weaknesses of FO specifications

Example:

$\psi$: The system has an infinite active run, along which it may always reach an inactive state.

$\psi$ cannot be expressed in FO.

We need quantifications on runs: $\psi = EG(Active \land EF \neg Active)$

- $E$: for some infinite run
- $A$: for all infinite runs
MSO Specifications

Definition: Syntax of MSO(AP, <)

\[ \varphi ::= \bot | p(x) | x = y | x < y | x \in X | \neg \varphi | \varphi \land \varphi | \exists x \varphi | \exists X \varphi \]

where \( p \in \text{AP}, \ x, y \) are first-order variables and \( X \) is a second-order variable.

Definition: Semantics of MSO(AP, <)

Let \( w = (T, <, \lambda) \) be a temporal structure over AP.

An assignment \( \nu \) maps first-order variables to time points in \( T \)

and second-order variables to sets of time points.

The semantics of first-order constructs is unchanged.

\[ w, \nu \models x \in X \quad \text{if} \quad \nu(x) \in \nu(X) \]

\[ w, \nu \models \exists X \varphi \quad \text{if} \quad w, \nu[X \mapsto T] \models \varphi \quad \text{for some} \ T \subseteq T \]

where \( \nu[X \mapsto T] \) maps \( X \) to \( T \) and keeps unchanged the other assignments.
MSO vs Temporal

MSO logic

- MSO(⟨⟩) has a good expressive power
  ... but MSO(⟨⟩)-formulae are not easy to write and to understand.
- MSO(⟨⟩) is decidable on computation trees
  ... but satisfiability and model checking are non elementary.

We need a temporal logic

- with no explicit variables,
- allowing quantifications over runs,
- usual specifications should be easy to write and read,
- with good complexity for satisfiability and model checking problems,
- with good expressive power.

Computation Tree Logic CTL* introduced by Emerson & Halpern (1986).
**CTL* (Emerson & Halpern 86)**

**Definition: Syntax of the Computation Tree Logic CTL*(AP, SU)**

\[ \varphi ::= \bot \mid p \ (p \in AP) \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \text{ SU } \varphi \mid E \varphi \mid A \varphi \]

We may also add the past modality SS. Two implicit free variables.

**Definition: Semantics of CTL*(AP, SU)**

Let \( M = (S, T, I, AP, \ell) \) be a Kripke structure (encodes the computation tree \( T \)).

Let \( \sigma = s_0 s_1 s_2 \cdots \) be an infinite run of \( M \) (infinite branch of \( T \)).

\( i \in \mathbb{N} \) (current position in the run \( \sigma \)).

\[ \begin{align*}
M, \sigma, i \models p & \quad \text{if} \quad p \in \ell(s_i) \\
M, \sigma, i \models \varphi \text{ SU } \psi & \quad \text{if} \quad \exists k > i, \ M, \sigma, k \models \psi \quad \text{and} \quad \forall i < j < k, \ M, \sigma, j \models \varphi \\
M, \sigma, i \models E \varphi & \quad \text{if} \quad M, \sigma', i \models \varphi \quad \text{for some} \quad \text{infinite run} \quad \sigma' \quad \text{such that} \quad \sigma'[i] = \sigma[i] \\
M, \sigma, i \models A \varphi & \quad \text{if} \quad M, \sigma', i \models \varphi \quad \text{for all} \quad \text{infinite runs} \quad \sigma' \quad \text{such that} \quad \sigma'[i] = \sigma[i]
\end{align*} \]

where \( \sigma[i] = s_0 \cdots s_i \).

**Remark:**

\( \sigma'[i] = \sigma[i] \) means that future is branching but past is not.
Example: Some specifications

- EF $\varphi$: $\varphi$ is possible (FO-definable on CT)
- AG $\varphi$: $\varphi$ is an invariant (FO-definable on CT)
- AF $\varphi$: $\varphi$ is unavoidable (not FO-definable on CT)
- EG $\varphi$: $\varphi$ holds globally along some path (not FO-definable on CT)

Remark: Some equivalences

- $A \varphi \equiv \neg E \neg \varphi$
- $E(\varphi \lor \psi) \equiv E \varphi \lor E \psi$
- $A(\varphi \land \psi) \equiv A \varphi \land A \psi$

Theorem: $CTL^* \subseteq MSO$

For each $\varphi \in CTL^*(AP, SU)$ we can construct an equivalent formula with two free variables $\tilde{\varphi}(X, x) \in MSO(AP, <)$.

For all computation tree $T$, infinite branch $B$ of $T$ and position $i$ in $B$, we have $T, B, i \models \varphi$ iff $T, X \mapsto B, x \mapsto i \models \tilde{\varphi}$.
Model checking of $\text{CTL}^*$

Definition: Existential and universal model checking

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

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

$M \models \forall \varphi$ if $M, \sigma, 0 \models \varphi$ for all initial infinite runs $\sigma$ of $M$.

Remark: $M \models \forall \varphi$ iff $M \not\models \exists \neg \varphi$

Remark: Often, formulas start with E or A and if $M$ has a single initial state, we do not need to distinguish between $\models \exists$ and $\models \forall$.

Definition: Model checking problems $\text{MC}_{\text{CTL}^*}^\forall$ and $\text{MC}_{\text{CTL}^*}^\exists$

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

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

Theorem:

The model checking problem for $\text{CTL}^*$ is PSPACE-complete. Proof later
**State formulae and path formulae**

**Definition: State formulae**

ϕ ∈ CTL* is a state formula if ∀M, σ, σ’, i, j such that σ(i) = σ'(j) we have

\[ M, σ, i \models ϕ \iff M, σ', j \models ϕ \]

If ϕ is a state formula and M = (S, T, I, AP, ℓ), define

\[ M, s \models ϕ \text{ if } M, σ, 0 \models ϕ \text{ for some infinite run } σ \text{ of } M \text{ with } σ(0) = s \]

and

\[ [ϕ]^M = \{ s ∈ S \mid M, s \models ϕ \} \]

**Example: State formulae**

Atomic propositions are state formulae: \[ [p] = \{ s ∈ S \mid p ∈ ℓ(s) \} \]

State formulae are closed under boolean connectives.

\[ [¬ϕ] = S \setminus [ϕ] \quad [ϕ_1 \lor ϕ_2] = [ϕ_1] \cup [ϕ_2] \]

Formulae of the form Eϕ or Aϕ are state formulae, provided ϕ is future.

Remark:

\[ M \models ∃ϕ \text{ iff } I \cap [Eϕ] \neq ∅ \quad M \models ∀ϕ \text{ iff } I \subseteq [Aϕ] \]

**Definition: Alternative syntax**

State formulae

\[ ϕ ::= ⊥ \mid p \ (p ∈ AP) \mid ¬ϕ \mid ϕ \lor ϕ \mid Eψ \mid Aψ \]

Path formulae

\[ ψ ::= ϕ \mid ¬ψ \mid ψ \lor ψ \mid ψ SU ψ \]
Definition: Computation Tree Logic CTL(AP, X, U)

Syntax:

\[ \varphi ::= \bot \mid p \ (p \in AP) \mid \neg \varphi \mid \varphi \lor \varphi \mid EX \varphi \mid AX \varphi \mid E \varphi U \varphi \mid A \varphi U \varphi \]

The semantics is inherited from CTL*.

Remark: E \varphi U \psi is not FO-definable on the computation tree.

Remark: All CTL formulae are state formulae

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

Examples: Macros

- EF \varphi = E \top U \varphi \quad \text{and} \quad AG \varphi = \neg EF \neg \varphi
- AF \varphi = A \top U \varphi \quad \text{and} \quad EG \varphi = \neg AF \neg \varphi
- AG(req \rightarrow EF grant)
- AG(req \rightarrow AF grant)
Definition: Semantics

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

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

Theorem: $\text{CTL} \subseteq \text{MSO}$

For each $\varphi \in \text{CTL}(AP, X, U)$ we can construct an equivalent formula with one free variable $\tilde{\varphi}(x) \in \text{MSO}(AP, <)$.

NB. Here models are computation trees.
CTL (Clarke & Emerson 81)

Example:

\[
\begin{align*}
[\text{EX } p] &= \{1, 2, 3, 4, 5, 6, 8\} \\
[\text{AX } p] &= \{2, 3, 5, 6, 7, 8\} \\
[\text{EF } p] &= \{1, 2, 3, 4, 5, 6, 7, 8\} \\
[\text{AF } p] &= \{2, 3, 5, 6, 7, 8\} \\
[\text{E } q \text{ U } r] &= \{1, 2, 3, 4, 5, 6, 8\} \\
[\text{A } q \text{ U } r] &= \{2, 3, 5, 6, 7, 8\}
\end{align*}
\]
Remark: Equivalent formulae

- $\text{AX} \varphi \equiv \neg \text{EX} \neg \varphi$, assuming no deadlocks
- $\neg (\varphi \cup \psi) \equiv \text{G} \neg \psi \lor (\neg \psi \cup (\neg \varphi \land \neg \psi))$
- $\text{A} \varphi \cup \psi \equiv \neg \text{EG} \neg \psi \land \neg \text{E}(\neg \psi \cup (\neg \varphi \land \neg \psi))$
- $\text{AG}(\text{req} \rightarrow \text{F grant}) \equiv \text{AG}(\text{req} \rightarrow \text{AF grant})$
- $\text{AGF} \varphi \equiv \text{AGAF} \varphi$
- $\text{EFG} \varphi \equiv \text{EFEG} \varphi$
- $\text{EGAF} \varphi \implies \text{EGF} \varphi \implies \text{EGEF} \varphi$
  
  but $M_1 \models \text{EGF} \varphi$, $M_1 \not\models \text{EGAF} \varphi$ and $M_2 \models \text{EGEF} \varphi$, $M_2 \not\models \text{EGF} \varphi$.

- $\text{EGAF} \varphi \not\equiv \text{EGF} \varphi \not\equiv \text{EGEF} \varphi$
- $\text{AFEG} \varphi \not\equiv \text{AFG} \varphi \not\equiv \text{AFAG} \varphi$
- $\text{EGEX} \varphi \not\equiv \text{EGX} \varphi \not\equiv \text{EGAX} \varphi$
Model checking of $\text{CTL}$

**Definition: Existential and universal model checking**

Let $M = (S, T, I, AP, \ell)$ be a Kripke structure and $\varphi \in \text{CTL}$ 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 $I \cap \llbracket \varphi \rrbracket \neq \emptyset$
- $M \models \forall \varphi$ iff $I \subseteq \llbracket \varphi \rrbracket$
- $M \models \forall \varphi$ iff $M \not\models \exists \neg \varphi$

**Definition: Model checking problems $\text{MC}_{\text{CTL}}^\forall$ and $\text{MC}_{\text{CTL}}^\exists$**

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

**Question:** Does $M \models \forall \varphi$? or Does $M \models \exists \varphi$?

**Theorem:**

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


References

*Specification and Verification using Temporal Logics.* 
In Modern applications of automata theory, IISc Research Monographs 2. 
http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php

*Temporal logic: mathematical foundations and computational aspects.* 

On the temporal analysis of fairness. 

Checking that finite state concurrent programs satisfy their linear specification.  

The complexity of propositional linear temporal logic. 