Weaknesses of linear behaviors

Example:

φ: Whenever p holds, it is possible to reach a state where q holds.

φ cannot be checked on linear behaviors.

We need to consider the computation-trees.

Remark: FO definable on the computation tree

$$\forall x \; (p(x) \rightarrow \exists y \; (x < y \wedge q(y)))$$

Weaknesses of FO specifications

Example:

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

ψ cannot be expressed in FO.

We need quantifications on runs: $\psi = \mathsf{EG}(\mathrm{Active} \wedge \mathsf{EF}\, \neg\mathrm{Active})$

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

MSO Specifications

Definition: Syntax of $\mathrm{MSO}(\mathrm{AP}, <)$

$$\varphi ::= \bot \mid p(x) \mid x = y \mid x < y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \exists X\, \varphi$$

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

Definition: Semantics of $\mathrm{MSO}(\mathrm{AP}, <)$

Let $w = (\mathbb{T}, <, \lambda)$ be a temporal structure over AP.
An assignment ν maps first-order variables to time points in $\mathbb{T}$
and second-order variables to sets of time points.

The semantics of first-order constructs is unchanged.

$$\begin{array}{lll} w, \nu \models x \in X & \text{if} & \nu(x) \in \nu(X) \\ w, \nu \models \exists X\, \varphi & \text{if} & w, \nu[X \mapsto T] \models \varphi \text{ for some } T \subseteq \mathbb{T} \end{array}$$

where $\nu[X \mapsto T]$ maps X to T and keeps unchanged the other assignments.

MSO vs Temporal

MSO logic

- $\mathrm{MSO}(<)$ has a good expressive power
  ... but $\mathrm{MSO}(<)$-formulae are not easy to write and to understand.
- $\mathrm{MSO}(<)$ 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 $\mathrm{CTL}^*(\mathrm{AP}, \mathsf{SU})$

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

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

Definition: Semantics of $\mathrm{CTL}^*(\mathrm{AP}, \mathsf{SU})$

Let $M = (S, T, I, \mathrm{AP}, \ell)$ be a Kripke structure (encodes the computation tree T).
Let $\sigma = s_0 s_1 s_2 \cdots$ be an infinte run of M (infinite branch of T).
$i \in \mathbb{N}$ (current position in the run σ).

$M, \sigma, i \models p$	if	$p \in \ell(s_i)$
$M, \sigma, i \models \varphi \,\mathsf{SU}\, \psi$	if	$\exists k > i,\ M, \sigma, k \models \psi$ and $\forall i < j < k,\ M, \sigma, j \models \varphi$
$M, \sigma, i \models \mathsf{E}\varphi$	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:

- $\sigma'[i] = \sigma[i]$ means that future is branching but past is not.

CTL^* (Emerson & Halpern 86)

Example: Some specifications

- EF φ: φ is possible — FO-definable on CT
- AG φ: φ is an invariant — FO-definable on CT
- AF φ: φ is unavoidable — not FO-definable on CT
- EG φ: φ holds globally along some path — not FO-definable on CT

Remark: Some equivalences

- $\mathsf{A}\,\varphi \equiv \neg\,\mathsf{E}\,\neg\varphi$
- $\mathsf{E}(\varphi \vee \psi) \equiv \mathsf{E}\,\varphi \vee \mathsf{E}\,\psi$
- $\mathsf{A}(\varphi \wedge \psi) \equiv \mathsf{A}\,\varphi \wedge \mathsf{A}\,\psi$

Theorem: $\mathrm{CTL}^* \subseteq \mathrm{MSO}$

For each $\varphi \in \mathrm{CTL}^*(\mathrm{AP}, \mathsf{SU})$ we can construct an equivalent formula with two free variables $\widetilde{\varphi}(X, x) \in \mathrm{MSO}(\mathrm{AP}, <)$.
For all computation tree T, infinite branch B of T and position i in B, we have

$$T, B, i \models \varphi \text{ iff } T, X \mapsto B, x \mapsto i \models \widetilde{\varphi}.$$

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 $M, \sigma, 0 \models \varphi$ for some initial infinite run σ of M.
$M \models_\forall \varphi$ if $M, \sigma, 0 \models \varphi$ for all initial infinite runs σ 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 $\mathrm{MC}^\forall_{\mathrm{CTL}^*}$ 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 $M \models_\exists \varphi$?

Theorem:

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

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

$M, s \models \varphi$ if $M, \sigma, 0 \models \varphi$ for some infinite run σ of M with $\sigma(0) = s$

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

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 $\mathsf{E}\,\varphi$ or $\mathsf{A}\,\varphi$ are state formulae, provided φ is future.

Remark: $M \models_\exists \varphi$ iff $I \cap [\![\mathsf{E}\,\varphi]\!] \neq \emptyset$ $\qquad$ $M \models_\forall \varphi$ iff $I \subseteq [\![\mathsf{A}\,\varphi]\!]$

Definition: Alternative syntax

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

Path formulae $\quad \psi ::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \psi \,\mathsf{SU}\, \psi$

CTL (Clarke & Emerson 81)

Definition: Computation Tree Logic $\mathrm{CTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$

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: $\mathsf{E}\,\varphi\,\mathsf{U}\,\psi$ is not FO-definable on the computation tree.

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

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 $s \in S$.

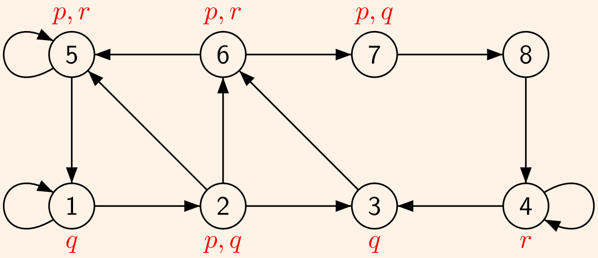
$$
\begin{array}{lll}
M, s \models p & \text{if} & p \in \ell(s) \\
M, s \models \mathsf{EX}\,\varphi & \text{if} & \exists s \to s' \text{ with } M, s' \models \varphi \\
M, s \models \mathsf{AX}\,\varphi & \text{if} & \forall s \to s' \text{ we have } M, s' \models \varphi \\
M, s \models \mathsf{E}\,\varphi\,\mathsf{U}\,\psi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

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

For each $\varphi \in \mathrm{CTL}(\mathrm{AP}, \mathsf{X}, \mathsf{U})$ we can construct an equivalent formula with one free variable $\widetilde{\varphi}(x) \in \mathrm{MSO}(\mathrm{AP}, <)$.
NB. Here models are computation trees.

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

- $\mathsf{AX}\,\varphi \equiv \neg\,\mathsf{EX}\,\neg\varphi$, assuming no deadlocks
- $\neg(\varphi\,\mathsf{U}\,\psi) \equiv \mathsf{G}\,\neg\psi \vee (\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$ discrete time
- $\mathsf{A}\,\varphi\,\mathsf{U}\,\psi \equiv \neg\,\mathsf{EG}\,\neg\psi \wedge \neg\,\mathsf{E}(\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi))$
- $\mathsf{AG}(\mathrm{req} \to \mathsf{F}\,\mathrm{grant}) \equiv \mathsf{AG}(\mathrm{req} \to \mathsf{AF}\,\mathrm{grant})$
- $\mathsf{A}\,\mathsf{G}\,\mathsf{F}\,\varphi \equiv \mathsf{AG}\,\mathsf{AF}\,\varphi$
- $\mathsf{E}\,\mathsf{F}\,\mathsf{G}\,\varphi \equiv \mathsf{EF}\,\mathsf{EG}\,\varphi$
- $\mathsf{EG}\,\mathsf{AF}\,\varphi \implies \mathsf{E}\,\mathsf{G}\,\mathsf{F}\,\varphi \implies \mathsf{EG}\,\mathsf{EF}\,\varphi$
  but $M_1 \models \mathsf{E}\,\mathsf{G}\,\mathsf{F}\,\varphi$, $M_1 \not\models \mathsf{EG}\,\mathsf{AF}\,\varphi$ and $M_2 \models \mathsf{EG}\,\mathsf{EF}\,\varphi$, $M_2 \not\models \mathsf{E}\,\mathsf{G}\,\mathsf{F}\,\varphi$.
- $\mathsf{EG}\,\mathsf{AF}\,\varphi \not\equiv \mathsf{E}\,\mathsf{G}\,\mathsf{F}\,\varphi \not\equiv \mathsf{EG}\,\mathsf{EF}\,\varphi$
- $\mathsf{AF}\,\mathsf{EG}\,\varphi \not\equiv \mathsf{A}\,\mathsf{F}\,\mathsf{G}\,\varphi \not\equiv \mathsf{AF}\,\mathsf{AG}\,\varphi$
- $\mathsf{EG}\,\mathsf{EX}\,\varphi \not\equiv \mathsf{E}\,\mathsf{G}\,\mathsf{X}\,\varphi \not\equiv \mathsf{EG}\,\mathsf{AX}\,\varphi$

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:

$$M \models_\exists \varphi \quad \text{iff} \quad I \cap [\![\varphi]\!] \neq \emptyset$$
$$M \models_\forall \varphi \quad \text{iff} \quad I \subseteq [\![\varphi]\!]$$
$$M \models_\forall \varphi \quad \text{iff} \quad M \not\models_\exists \neg\varphi$$

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 $M \models_\exists \varphi$?

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 $\mathcal{O}(|M| \cdot |\varphi|)$

References

[1] Christel Baier and Joost-Pieter Katoen.
Principles of Model Checking.
MIT Press, 2008.

[2] B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, Ph. Schnoebelen.
Systems and Software Verification. Model-Checking Techniques and Tools.
Springer, 2001.

[3] E.M. Clarke, O. Grumberg, D.A. Peled.
Model Checking.
MIT Press, 1999.

[4] Z. Manna and A. Pnueli.
The Temporal Logic of Reactive and Concurrent Systems: Specification.
Springer, 1991.

[5] Z. Manna and A. Pnueli.
Temporal Verification of Reactive Systems: Safety.
Springer, 1995.

[6] Ph. Schnoebelen.
The Complexity of Temporal Logic Model Checking.
In *AiML'02*, 393–436. King's College Publication, 2003.

References

[7] S. Demri and P. Gastin.
Specification and Verification using Temporal Logics.
In Modern applications of automata theory, IISc Research Monographs 2.
World Scientific, 2012.
`http://www.lsv.ens-cachan.fr/~gastin/mes-publis.php`

[8] D. Gabbay, I. Hodkinson and M. Reynolds.
Temporal logic: mathematical foundations and computational aspects.
Vol 1, Clarendon Press, Oxford, 1994.

[9] D. Gabbay, A. Pnueli, S. Shelah, and J. Stavi.
On the temporal analysis of fairness.
In *7th Annual ACM Symposium PoPL'80*, 163–173. ACM Press.

[10] O. Lichtenstein and A. Pnueli.
Checking that finite state concurrent programs satisfy their linear specification.
In *ACM Symposium PoPL'85*, 97–107.

[11] A. Sistla and E. Clarke.
The complexity of propositional linear temporal logic.
Journal of the Association for Computing Machinery. **32** (3), 733–749, (1985).