Temporal Abstract Interpretation

Patrick COUSOT and **Radhia COUSOT**

DI, École normale supérieure
45 rue d'Ulm
75230 Paris cedex 05, France
mailto:Patrick.Cousot@ens.fr
http://www.di.ens.fr/~cousot

LIX
École polytechnique & CNRS
91128 Palaiseau cedex, France
mailto:rcousot@lix.polytechnique.fr
http://lix.polytechnique.fr/~rcousot

27th ACM Principles of Programming Languages
Boston, MA, USA Wednesday January 19th, 2000

To have a continuum of program analysis techniques ranging from model-checking to static analysis.

1. Objective

Model-checking versus static analysis

- Both model-checking and static analysis are sound;
- Model-checking is seemingly complete (whereas static analysis is not);
- Abstract interpretation is useful to understand the approximations which are involved in both cases and to generalize;
- Useful since present-day abstract model-checking is not general enough: e.g. state-to-state abstraction does not fit for polyhedral model-checking.

What is in the paper?

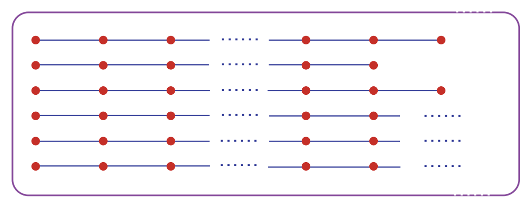
- We introduce a new temporal calculus, the reversible $\widehat{\mu}^\star$-calculus (generalizing known calculi/logics);
- We study its abstract interpretation (in a very general setting i.e. for any semantics and (co-)abstraction);
- Surprisingly, we show that its model-checking abstraction is incomplete (even for finite state models);
- We study sufficient completeness conditions (e.g. the CTL subcalculus is complete but not CTL*);
- We consider applications to abstract model checking and dataflow analysis.

What is in this talk?

- A few intuitive ideas to help read the paper.

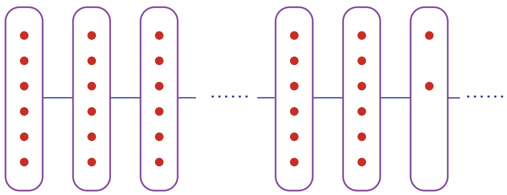
2. Abstract interpretation: abstraction/ concretization

An example of abstraction: a set of sequences of states



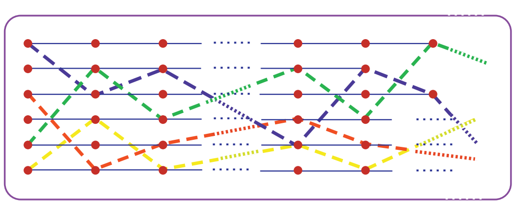
can be abstracted/approximated by .../...

An example of abstraction (cont'd)
a sequence of sets of states



Concretization

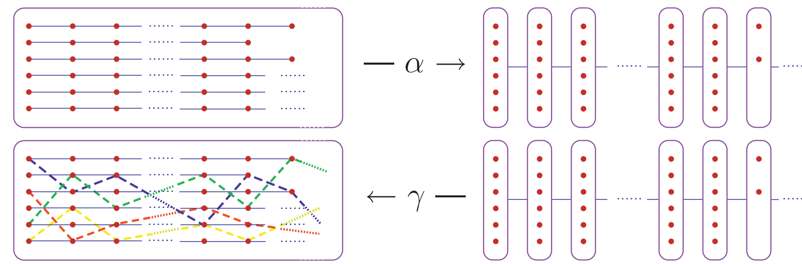
- The concretization contains all original traces:



plus (unrealistic) additional ones (- - -, - - -, - - -, - - -, …);
- Approximation from above (more traces than possible);
- The additional traces would yield the same abstraction anyway!

Set-based abstraction

Let us call this abstraction the set-based abstraction:



Abstraction … in general

- Abstraction can also be understood as choosing an abstract world as a subset of the concrete world (more precisely as a Moore family). Then:
  - The expressible concrete properties are closed/invariant under the abstraction so can be stated exactly in the abstract world;
  - The inexpressible concrete properties have to be upper- or lower-approximated by (preferably the best possible) abstract property;
- The abstract world is closed under join, meet, fixpoints, etc.

3. Temporal logics/calculi involve implicit abstractions

Implicit temporal abstractions

- In general, temporal-logic/calculi cannot express <u>all</u> properties of models, but only specific ones (e.g. [1]);
- The semantics of the temporal-logic/calculus can be understood as an abstraction of the concrete semantics (arbitrary sets of sequences of states);
- For example Kozen's propositional μ-calculus is closed for the set-based abstraction.

Reference

[1] Emerson, E. & Halpern, J. "Sometimes" and "Not Never" revisited: On branching time versus linear time. *TOPLAS 33* (1986), 151–178.

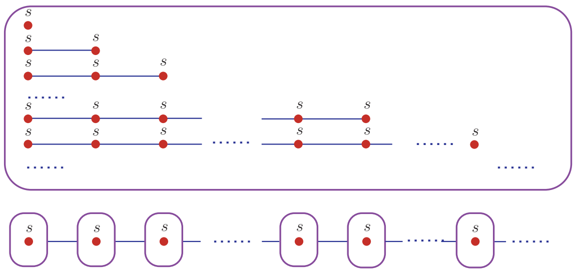
4. Abstract interpretation: soundness/ completeness

Intuition for soundness

For a *given class* of properties, soundness means that:

- Any property (in the *given class*) of the abstract world must hold in the concrete world;
- For the set-based abstraction:
  - Example: "on any trace, state a can never be immediately followed by state b";
  - Counter-Example: "all traces are infinite";

Example for unsoundness



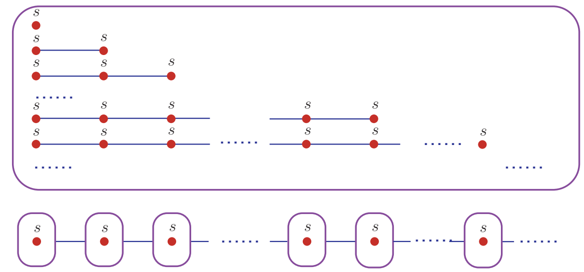
All abstract traces are infinite but not the concrete ones!

Intuition for completeness

For a *given class* of properties, completeness means that:

- Any property (in the *given class*) of the concrete world must hold in the abstract world;
- For the set-based abstraction:
  - Example: "execution from state a must eventually be followed by states b or c";
  - Counter-Example: "all traces are finite";

Example for incompleteness



All concrete traces are finite but not the abstract ones!

5. Model/checking is an abstract interpretation

Model-checking

- *Universal model-checking* checks that:

$$\text{Model} \subseteq \text{Temporal specification}$$

- Less frequently, we also have the dual *existential model-checking*:

$$\text{Model} \cap \text{Temporal specification} \neq \emptyset$$

Model-checking is a boolean abstraction

- Knowing only whether or not "a specification φ is satisfied by all traces of a model M" is a boolean abstraction (a loss of information):

$$\alpha^{\forall}_{M}(\varphi) \stackrel{\triangle}{=} (M \subseteq \varphi)$$

- The concretization is the model satisfying the specification:

$$\gamma^{\forall}_{M}(\mathrm{ff}) \stackrel{\triangle}{=} \emptyset$$
$$\gamma^{\forall}_{M}(\mathrm{tt}) \stackrel{\triangle}{=} M$$

- Universal model-checking is a Galois connection:

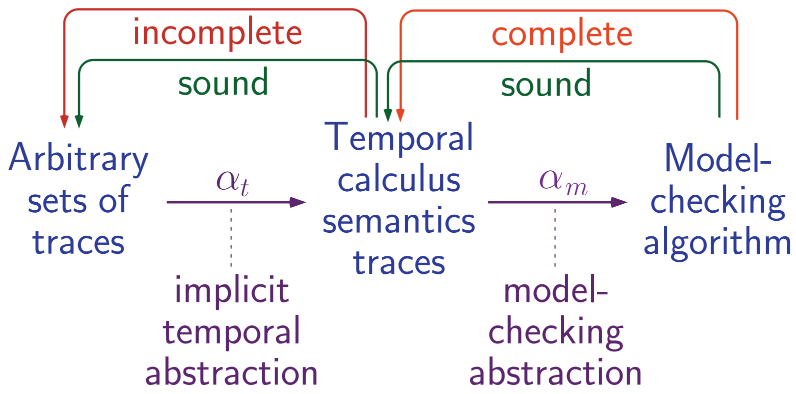
$$\langle \text{Sets of traces}, \supseteq \rangle \underset{\alpha^{\forall}_{M}}{\overset{\gamma^{\forall}_{M}}{\leftrightarrows}} \langle \{\mathrm{ff}, \mathrm{tt}\}, \Longleftarrow \rangle$$

- Dually, existential model-checking is also a Galois connection;
- In abstract interpretation theory, Galois connections formalize the notion of discrete approximation;
- The model-checking algorithms can be constructively derived by abstract interpretation of the temporal logic/calculus semantics.

Relative completeness

- The completeness result for the model-checking abstraction is relative to the semantics of the temporal logic/calculus!
- So completeness is relative to the abstract world of the temporal logic/calculus semantics not to the concrete world of arbitrary sets of traces!
- This implicit abstraction is itself incomplete (e.g. for the reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus, even for finite state models);
- Intuition: with general temporal specifications, model-checking algorithms cannot deal with sets of states only and would have to handle sets of traces (too costly).

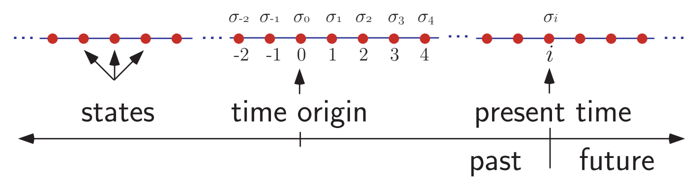
Relative completeness



6. A few words on the reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus

Semantic domain of the reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus

- The semantics of a formula of the reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus is a set of infinite time-symmetric traces;
- An infinite time-symmetric trace $\langle i,\ \sigma\rangle$:

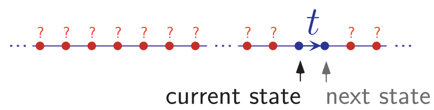


The reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus

φ ::=	$\boldsymbol{\sigma}_S$	$S \in \wp(\mathbb{S})$	state predicate
|	$\boldsymbol{\pi}_t$	$t \in \wp(\mathbb{S} \times \mathbb{S})$	transition predicate
|	$\oplus\,\varphi_1$		next
|	$\varphi_1^{\curvearrowleft}$		reversal
|	$\varphi_1 \vee \varphi_2$		disjunction
|	$\neg\,\varphi_1$		negation
|	X	$X \in \mathbb{X}$	variable
|	$\boldsymbol{\mu}\,X \cdot \varphi_1$		least fixpoint
|	$\boldsymbol{\nu}\,X \cdot \varphi_1$		greatest fixpoint
|	$\forall\,\varphi_1 : \varphi_2$		universal state closure

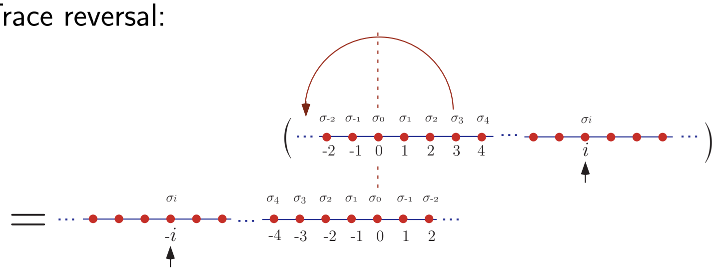
Transition predicates π_t

- The transition predicate π_t denotes all traces with a transition t from current to next state:



Reversal $^\curvearrowleft$

- Trace reversal:



- Model reversal:

$$M^\curvearrowleft \triangleq \{\langle i,\ \sigma\rangle \mid \langle i,\ \sigma\rangle^\curvearrowleft \in M\}$$

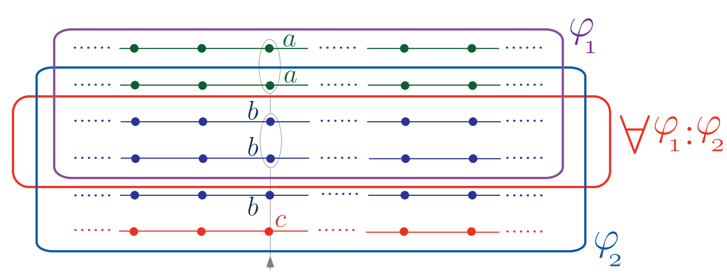
Abbreviations (examples)

$\varphi_1 \,\mathbf{U}\, \varphi_2 \triangleq \boldsymbol{\mu} X \cdot (\varphi_2 \vee (\varphi_1 \wedge \oplus X))$ until

$\varphi_1 \,\mathbf{S}\, \varphi_2 \triangleq (\varphi_1{}^\curvearrowleft \,\mathbf{U}\, \varphi_2{}^\curvearrowleft)^\curvearrowleft$ since

Universal state closure

- The universal state closure $\forall \varphi_1 : \varphi_2$ is the set of traces of φ_1 such that all traces in φ_1 with the same current state belong to φ_2;



Subcalculi (example: Kozen's propositional μ-calculus)

$$\varphi \ ::= \ \boldsymbol{\sigma}_S \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg\, \varphi_1 \mid \Box\, \varphi_1 \mid \Diamond\, \varphi_1 \mid X \mid \boldsymbol{\mu}\, X \cdot \varphi_1 \mid \boldsymbol{\nu}\, X \cdot \varphi_1$$

where:

τ : transition relation (program SOS semantics);

$\Box\, \varphi_1 \stackrel{\triangle}{=} \forall\, \boldsymbol{\pi}_\tau : \oplus\, \varphi_1$ always (after next step);

$\Diamond\, \varphi_1 \stackrel{\triangle}{=} \exists\, \boldsymbol{\pi}_\tau : \oplus\, \varphi_1$ sometime (after next step).

On the reversible $\overset{\curvearrowleft}{\mu}^{\star}$-calculus

- Generalization of previous temporal logics and calculi;
- Contrary to previous propositions:
  - Every logical statement is explicit (e.g. no implicit underlying Kripke structure),
  - A single temporal operator $\curvearrowleft$ to handle past and future,
  - Completely time-symmetric,
  - Model-checking of the full calculus is incomplete (complete for subcalculi e.g. CTL versus CTL*.

7. Conclusion

More in the paper ...

- Compositional abstract interpretation of generic μ-calculi (independently of a particular semantics, including for non-monotone operators);
- Study of the model-checking abstractions;
- Study of (sufficient) abstraction completeness conditions;
- Identification of model-checking complete subcalculi;
- Applications to:
  - Abstract model checking;
  - Dataflow analysis (and the soundness of live variables).

Perspectives

- Model-checking is an incomplete abstract interpretation;
- So for infinite state systems and more general temporal logics:
  - other abstractions can be used (e.g. not boolean, not state-to-state, as in abstract testing);
  - because of incompleteness, the usual model-checking algorithms are not the most precise possible ones, so other algorithms should be used [1].

Reference

[1] P. Cousot and R. Cousot. Abstract interpretation and application to logic programs. *Journal of Logic Programming*, 13(2–3):103–179, 1992.

The End