1. Objective

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

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 $\mu^\ast$-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$^\ast$);
- 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:
- 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

- In general, temporal-logic/calculi cannot express all 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 $\mu$-calculus is closed for the set-based abstraction.

References

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”;

References
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”;

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 \( \varphi \) is satisfied by all traces of a model \( M \)" is a boolean abstraction (a loss of information):
  \[ \alpha^\varphi_M(\varphi) \triangleq (M \subseteq \varphi) \]

• The **concretization** is the model satisfying the specification:
  \[ \begin{align*}
  \gamma^\varphi_M(\text{ff}) & \triangleq \emptyset \\
  \gamma^\varphi_M(\text{tt}) & \triangleq M
  \end{align*} \]

• **Universal model-checking** is a Galois connection:
  \[ \langle \text{Sets of traces, } \supseteq \rangle \xrightarrow{\alpha^\varphi_M} \langle \{\text{ff, tt}\}, \iff \rangle \]

  • 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 \( \mu \)-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

- Temporal calculus semantics traces
  - Implicit temporal abstraction
  - Model-checking algorithm

Arbitrary sets of traces

\[ \alpha_t \]

Complete sound

\[ \alpha_m \]

Semantics domain of the reversible \( \mu^\ast \)-calculus

- The semantics of a formula of the reversible \( \mu^\ast \)-calculus is a set of infinite time-symmetric traces;
- An infinite time-symmetric trace \( \langle i, \sigma \rangle \):

\[
\begin{array}{cccccccc}
\ldots & \sigma_{-2} & \sigma_{-1} & \sigma_0 & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \ldots \\
\text{states} & \text{time origin} & -2 & -1 & 0 & 1 & 2 & 3 & 4 & \text{present time} \\
\text{past} & \text{future} & \downarrow & \downarrow & \uparrow & \uparrow & \downarrow & \downarrow & \uparrow & \uparrow \\
\end{array}
\]

6. A few words on the reversible \( \mu^\ast \)-calculus

\( \varphi ::= \sigma_S \quad S \in \wp(S) \)  
state predicate

\( | \pi_t \quad t \in \wp(S \times S) \)  
transition predicate

\( | \oplus \varphi_1 \)  
next

\( | \varphi_1 \)  
reversal

\( | \varphi_1 \lor \varphi_2 \)  
disjunction

\( | \neg \varphi_1 \)  
negation

\( | X \quad X \in X \)  
variable

\( | \mu X \cdot \varphi_1 \)  
least fixpoint

\( | \nu X \cdot \varphi_1 \)  
greatest fixpoint

\( | \forall \varphi_1 : \varphi_2 \)  
universal state closure
**Transition predicates $\pi_t$**

- The *transition predicate* $\pi_t$ denotes all traces with a transition $t$ from current to next state:

  ![Diagram](image)

  - Current state
  - Next state

**Abbreviations (examples)**

- $\varphi_1 \cup \varphi_2 \triangleq \mu X \cdot (\varphi_2 \lor (\varphi_1 \land \oplus X))$ until
- $\varphi_1 \mathcal{S} \varphi_2 \triangleq (\varphi_1 \cap \varphi_2) \cap$ since

**Reversal $\triangleleft$**

- Trace reversal:

  ![Diagram](image)

- Model reversal:

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

**Universal state closure**

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

  ![Diagram](image)
Subcalculi
(example: Kozen’s propositional $\mu$-calculus)

$\varphi ::= \sigma_S \mid \varphi_1 \lor \varphi_2 \mid \varphi_1 \land \varphi_2 \mid \neg \varphi_1 \mid \square \varphi_1 \mid \lozenge \varphi_1$
$X \mid \mu X \cdot \varphi_1 \mid \nu X \cdot \varphi_1$

where:
- $\tau$ : transition relation (program SOS semantics);
- $\square \varphi_1 \triangleq \forall \pi : \tau \cdot \varphi_1$ always (after next step);
- $\lozenge \varphi_1 \triangleq \exists \pi : \tau \cdot \varphi_1$ sometime (after next step).

On the reversible $\mu^\circ$-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 $\ominus$ 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 $\mu$-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


© P. Cousot & R. Cousot — POPL’00, January 19th, 2000

The End