Model Checking for the Practical Verificationist: A User’s Perspective on SAL

Lee Pike
leepike@galois.com

Galois Inc.

November 6, 2007
Goals (and non-Goals)

Goals:
- “Show-off” novel and/or useful language feature or tools.
- Begin a dialogue with other SAL users.

non-Goals:
- Provide a full SAL tutorial.
- Compare & contrast SAL to other model checkers.

…I could imagine a “SAL cookbook” or sets of libraries on the wiki being very useful.
Outline

- *Practical invariants*
- Higher-order functions

Not covered (but in the paper):
- Temporal refinement in SAL
- Environmental constraints
- Model checking + theorem-proving
  - Counterexample discovery
  - FMCAD’07 paper...
Cheap Invariants

- **k-Induction** to strengthen invariants *automatically*.
  - Generalizes induction over transition systems.
  - Automatic, but exponential in the size of $k$.
- **Disjunctive invariants**.
  - Each disjunction covers some configuration of the system.
  - Developed by Pnueli & Rushby, independently.
  - A disjunctive invariant can be built iteratively to cover the reachable states from the counterexamples returned by SAL for the hypothesized invariant being verified.

Okay, onto a definition of $k$-induction and two examples...
$k$-induction

Generalize from single transitions to trajectories of fixed length.

Consider a transition system $\langle S, S^0, \rightarrow \rangle$. For safety property $P$, show

- **Base**: If $s_0 \in S^0$, then for all trajectories $s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_k$, $P(s_i)$ for $0 \leq i \leq k$;
- **IS**: For all trajectories $s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_k$, if $P(s_i)$ for $0 \leq i \leq k - 1$, then $P(s_k)$.

Conclude that for all reachable $s$, $P(s)$.

Induction is the special case when $k = 1$. 
A SAL interlude...
SAL wishlist

- A type-checker that returns type-correctness conditions.
- More/better documentation for using the API.
- More tools for running proofs, outputting proofs, pretty-printing counterexamples, etc.
  I’ve written a few I’ll try to release.
Two workshop bold claims:

- SAL/Yices obviates the need for specialized real-time model checkers.
- SAL/Yices will make the need for full mechanical theorem-proving obsolete in many domains.
Thanks!

- **SAL coauthors**: Geoffrey Brown, Paul Miner, Steve Johnson, and Wilfredo Torres-Pomales.
- **Comments**: Levent Erkök at Galois, Inc. and the workshop reviewers.
Web resources

Slides, specifications, and proofs
Google: lee pike