Proof by Verified Symbolic Execution in ACL2

Sol Swords
sswords@cs.utexas.edu

May 9, 2009

G in the Logic

Recall Boyer and Hunt's G system:

- Symbolic functions in raw Lisp
- Symbolic data objects are in raw Lisp, unrepresentable in ACL2
- Usable for proof with unverified clause processor

G in the Logic, or "GL":

- Symbolic functions are in the ACL2 logic
- Symbolic objects are a subset of ACL2 objects
- Automatically-proven theorems about the system allow proof without trusting anything but ACL2 itself

The GL System

$$\begin{tikzcd}
5 \arrow[rr, bend left] & P \arrow[d, mapsto, "\textit{Code Transform}"] & \textit{false} \\
\boxed{A_2}\boxed{A_1}\boxed{A_0} \arrow[u, bend left, "ev_s(\_,env)"] \arrow[rr, bend right] & P_{sym} & \neg\,(A_2 \bigvee A_1 \bigvee A_0\,) \arrow[u, bend right, "ev_s(\_,env)"']
\end{tikzcd}$$

$$env = \{A_2 = 1,\; A_1 = 0,\; A_0 = 1\}$$

- Code transform produces symbolic simulator P_{sym} given program P
- Symbolic object semantics defined by evaluator ev_s
- Correspondence proof automatically generated for P_{sym}