What is *Sugar 2.0*?

- *Sugar 2.0* is the property language selected as a standard by *Accellera*

  Accellera was formed in 2000 through the unification of Open Verilog International and VHDL International to focus on identifying new standards, development of standards and formats, and to foster the adoption of new methodologies.

  Accellera's mission is to drive worldwide development and use of standards required by systems, semiconductor and design tools companies, which enhance a language-based design automation process.

- *Accellera* competition finalists
  - Motorola's CBV language
  - IBM's Sugar
  - Intel's ForSpec
  - Verisity's *e* language

From *Sugar 1* to *Sugar 2.0*

- *Sugar 1* is the property language of IBM's RuleBase model checking tool
  - based on CTL and regular expressions (ITL)
  - conventional textbook semantics based on infinite paths
- *Sugar 2.0* is *Sugar 1* plus features required by *Accellera*
  - based on LTL and regular expressions (but still has 'optional' CTL)
  - complex semantics for both finite and infinite paths
    - infinite path semantics for model checking
    - finite path semantics for checking properties on simulation runs
  - clocking constructs: multiple clocks with strong and weak clocking
    - $f\texttt{@}clk!$ evaluate f with respect to clock clk, assume clock live
    - $f\texttt{@}clk$ evaluate f with respect to clock clk, not necessarily live

Embedding *Sugar 2.0* in higher order logic

- Deep semantic embedding
  - datatypes in logic for Sugar constructs
    - boolean and regular expressions, LTL and CTL formulas
  - semantics represented in higher order logic
    - straightforward encoding of official Kripke structure semantics
- Analysing Sugar using the HOL system
  - proved simple ‘sanity checking’ lemmas
    - revealed minor errors in official semantics
  - working on more difficult proofs
    - correctness of clock elimination rewrites
  - Deriving rules
    - can combine temporal logic and HOL

On the poster

- Example *Sugar 2.0* constructs and definitional extensions
- Official semantics and HOL semantics of an LTL formula
- Typical examples of minor bugs found by trying to prove properties
- Two easily derived rules illustrating ‘added value’ of embedding in HOL