Orc in ACL2

Chad Wellington
9/05/2007
Orc, in general

- Assumptions
  - Distributed
  - “Workflow”
  - “Formal” Specification

- Composition
  - Sites
  - Combinators
Intuition, briefly

- Sites a form of Remote Procedure Call
  - publishes 0 or 1 times
- Connectives
  - bar: “f | g”
    - simultaneous, non-communicating parallel
  - pipe: “f >x> g”
    - sequential creation of g's on f's publications
    - analogy to dataflow, Unix pipe
  - where: “f where x<-g” or more recently “f<x<g”
    - parallel execution with binding and termination
    - fork-join parallelism
Operational Semantics

- Formalize program meaning
  - analogy to eval (?)
  - as transitions, not final result
- State transition definitions
  - conditional clauses (like Horn clauses)
  - can form chain (or tree)
  - here, labeled
Site definition

- **actuals**
  - \( x :: \text{unbound vbl} \)
  - \( v :: \text{bound vbl / value} \)

- **Sites**
  - \( M(x) :: \text{blocked} \)
  - \( M(v) :: \text{unblocked} \)
  - \( k :: \text{unique handle} \)
  - \( 0 :: \text{silent} \)
  - \( \text{let}(x) :: \text{publication} \)
Life cycle of a site

\[ M(x) \rightarrow^* M(v) \rightarrow ?k \rightarrow \text{let}(v) \rightarrow 0 \]
Publication rules

\[
f \xrightarrow{\tau} f'
\]
\[
f > x > g \xrightarrow{\tau} (f' > x > g) \mid [v/x].g
\]

(SEQ1V)

\[
g \xrightarrow{\tau} g'
\]
\[
f \text{ where } x : \in g \xrightarrow{\tau} [v/x].f
\]

(ASYM1V)

• “tau” is an internal event

• \([v/x].g = \) all free occurrences of \(x\) in \(g\) replaced by \(v\) (relaxed to all occurrences since we can rename variables)
Transition passing rules

\[
\frac{f \xrightarrow{a} f'}{g \xrightarrow{a} g'} \quad \text{(SYM1)}
\]

\[
\frac{f \xrightarrow{a} f'}{f \mid g \xrightarrow{a} f' \mid g} \quad \text{(ASYM1N)}
\]

\[
\frac{g \xrightarrow{a} g'}{f \mid g \xrightarrow{a} f \mid g'} \quad \text{(SYM2)}
\]

\[
\frac{\text{f where } x \in g \xrightarrow{a} f \text{ where } x \in g'}{\text{f where } x \in g \xrightarrow{a} f\text{ where } x \in g'} \quad \text{(ASYM2)}
\]

\[
\frac{\text{f where } x \in g \xrightarrow{a} f\text{ where } x \in g'}{\text{f where } x \in g \xrightarrow{a} f\text{ where } x \in g'} \quad \text{(SEQ1N)}
\]

- Upper half pass any transition
- Lower half exempt publications
&c.

- structure forces a sequence, not a tree
  - identifiable sequence of steps = “execution”
  - eliminate tau's = “trace”
    - note origin of tau's
- I make convenient (necessary?) assumptions
  - round-based execution
  - currently model semantic steps of tau transitions
    - others execute in separate segment of the round, need different treatment
- Code on web, documentation available but shaky