Semantics of Orc Logical Time

by Adrian Quark

May 8, 2009

1 Introduction

We describe a proposed semantics for nested logical clocks in Orc. This semantics is an extension of the asynchronous semantics presented in: D. Kitchin, W. R. Cook, and J. Misra. A language for task orchestration and its semantic properties. In *CONCUR*, pages 477–491, 2006.

2 Site Calls

The SOS semantics of Kitchin et al does not address the issue of how and when site return values are made available. To remedy this, we lift the reduction relation to operate over pairs $\langle r,\ f\rangle$, where r is a list of site return events which are available and f is an expression. It is understood that the environment will provide reductions of the form $\langle r,\ f\rangle \to \langle \{k?v\} \cup r,\ f\rangle$ to introduce available site returns.

Reduction rules of the form $\frac{f \to f' \quad \ldots}{g \to g'}$ are understood as $\frac{\langle r,\ f\rangle \to \langle r',\ f'\rangle \quad \ldots}{\langle r,\ g\rangle \to \langle r',\ g'\rangle}$.
SiteRet is redefined to consume an available site return event.

$$\frac{}{\langle \{k?v\} \cup r, ?k\rangle \xrightarrow{k?v} \langle r, \mathrm{let}(v)\rangle}$$

All other reductions of the form $f \to f'$ are understood as $\langle \emptyset, f\rangle \to \langle \emptyset, f'\rangle$.

3 Clocks

We introduce a new expression form $\textsc{Clock}(f, t, S)$ representing a logical clock. f is the clock's body expression, t is the clock's current time, and S is the clock's set of future events, of the form (k, t) where k is a site call handle and t is a time. Initially, all Clock expressions have the form $\textsc{Clock}(f, 0, \{\})$.

4 Quiescence

We introduce a new predicate qu on expressions. $\mathrm{qu}(f)$ holds iff f is quiescent. A quiescent expression cannot make any progress on its own. In essense, qu is the complement of the reduction relation.

A halted expression is quiescent.

$$\frac{}{\mathrm{qu}(\texttt{stop})}$$

A parallel composition is quiescent if its components are.

$$\frac{\mathrm{qu}(f) \quad \mathrm{qu}(g)}{\mathrm{qu}(f \ \mid\ g)}$$

A sequential composition is quiescent if its LHS is.

$$\frac{\mathrm{qu}(f)}{\mathrm{qu}(f \ \texttt{>}x\texttt{>}\ g)}$$

A pruning composition is quiescent if its components are.

$$\frac{\mathrm{qu}(f) \quad \mathrm{qu}(g)}{\mathrm{qu}(f \ \texttt{<}x\texttt{<} \ g)}$$

An otherwise composition is quiescent if its LHS is.

$$\frac{\mathrm{qu}(f)}{\mathrm{qu}(f \ \texttt{;} \ g)}$$

An unbound variable is quiescent.

$$\frac{}{\mathrm{qu}(M(x))}$$

A clock is quiescent if it has no future events and its body is quiescent.

$$\frac{\mathrm{qu}(f)}{\mathrm{qu}(\textsc{Clock}(f, t, \{\}))}$$

A pending call to Ltimer is quiescent.

$$\frac{f \stackrel{\texttt{Ltimer}_k(v)}{\to} f'}{\mathrm{qu}(?k)}$$

In general, $\mathrm{qu}(?k)$ must be defined by the environment for a given k based on the site called, arguments, and global state. Some site calls may change quiescence before returning. In that case, the environment must provide rules of the form $\mathrm{qu}(?k_q)$, $?k \stackrel{\tau}{\to} ?k_q$, $?k_q \stackrel{\tau}{\to} ?k_{q'}$, and $?k_q \stackrel{k?v}{\to} \mathrm{let}(v)$ in order to track the internal state q of the site call.

Note that an expression may change from quiescent to non-quiescent in reaction to external events. For example, in the program `c.get() | c.put(signal)`, the pending call `c.get()` remains quiescent until `c.put(signal)` is evaluated, at which point `c.get()` becomes non-quiescent.

5 Small-Step Asynchronous Semantics

We extend the small-step asynchronous semantics to deal with clock expressions.

When a clock's body calls Ltimer, the clock records a future event. Note that the call to Ltimer is not visible outside clock.

$$\frac{f \stackrel{\texttt{Ltimer}_k(v)}{\to} f'}{\textsc{Clock}(f, t, S) \stackrel{\tau}{\to} \textsc{Clock}(f', t, \{(k, t+v)\} \cup S)}$$

The clock expression is transparent to all other transitions of its body.

$$\frac{f \stackrel{a}{\to} f' \quad a \neq \texttt{Ltimer}_k(v)}{\textsc{Clock}(f, t, S) \stackrel{a}{\to} \textsc{Clock}(f', t, S)}$$

When a clock's body is quiescent, the clock advances to the next time step and all pending site calls waiting for that time step return in arbitrary order.

$$\frac{\mathrm{qu}(f_0) \quad \forall (k_i, t_i') \in S.\ t_i' > t_0' \quad \forall k_i \in \{k_1, \ldots, k_n\}.\ \langle \{k_i?\texttt{signal}\}, f_{i-1} \rangle \stackrel{k_i?\texttt{signal}}{\to} \langle \emptyset, f_i \rangle}{\textsc{Clock}(f_0, t, \{(k_1, t_0'), \ldots, (k_n, t_0')\} \cup S)) \stackrel{\tau}{\to} \textsc{Clock}(f_n, t_0', S)}$$