CS 524 Lecture 12b
José Meseguer

A Partial Order of Events Model for Object-Based Rewrite Theories

The ideas presented in this lecture are based on the paper:

J. Meseguer and C. Talcott, "A Partial Order Event Model for Concurrent Objects," Proc. CONCUR'99, Springer LNCS 1664, 415-430, 1999.

We already saw that for Petri nets we can have:

1. An algebraic model of its concurrent computations, namely the category $\mathcal{T}_{\mathcal{N}}$ of the net $\mathcal{N}$.
2. A topological model of its concurrent computations, namely the Best-Devillers processes $\mathcal{P}_{\mathcal{N}}$ of $\mathcal{N}$,

and that these two models coincide, i.e., they are isomorphic up to a change of representation.

The advantages of having both (1) and (2) is that one can have one's cake and eat it too, i.e., one can both:

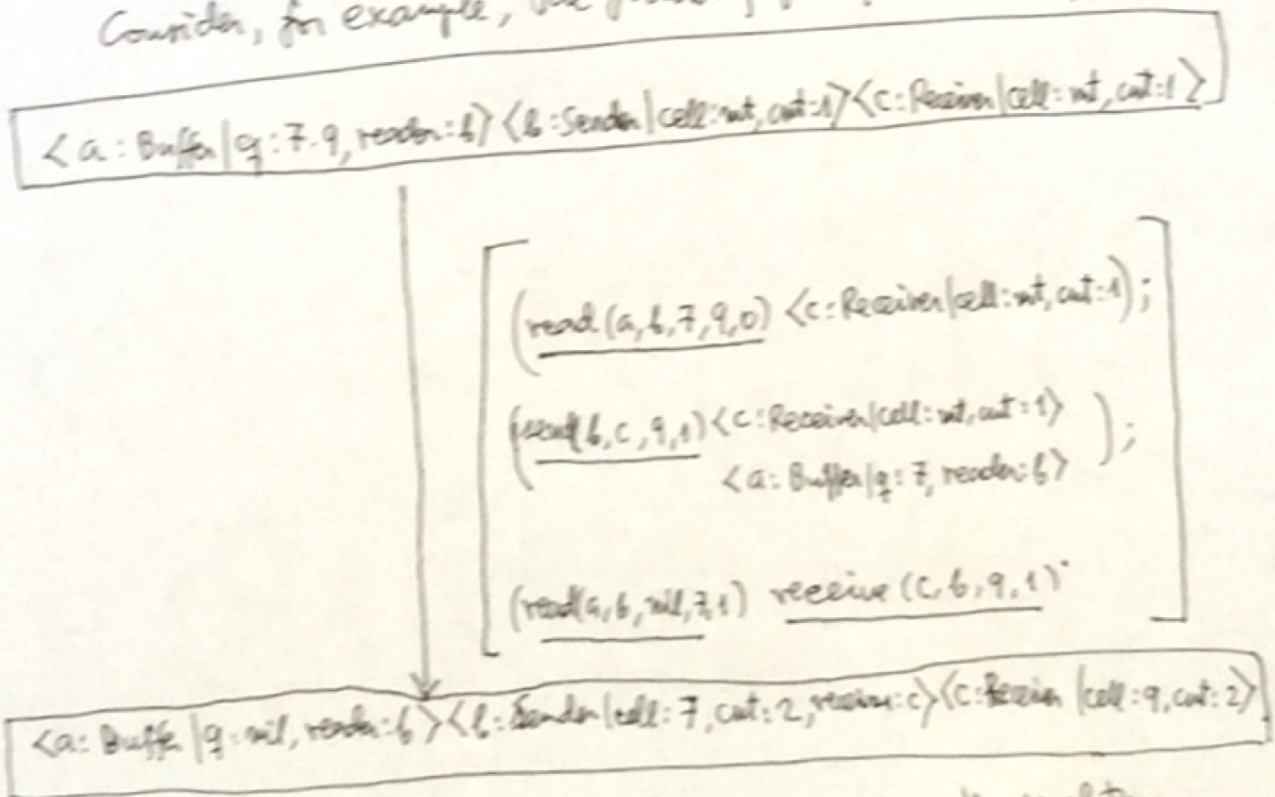
a. have an intuitive, pictorial model of concurrent computations, and
b. reason mathematically about such computations in an algebraic way

Since concurrent object systems are probably the most widely applicable model of concurrent computation and the best suited to describe distributed algorithms and systems,

Wouldn't it be nice if we can also get a "topological", pictorial model of their computations?

Yes, it would. But how would it look like? We can get a glimpse by considering a concrete example of our sender, receiver and buffer object system.

Consider, for example, the following proof term computation:



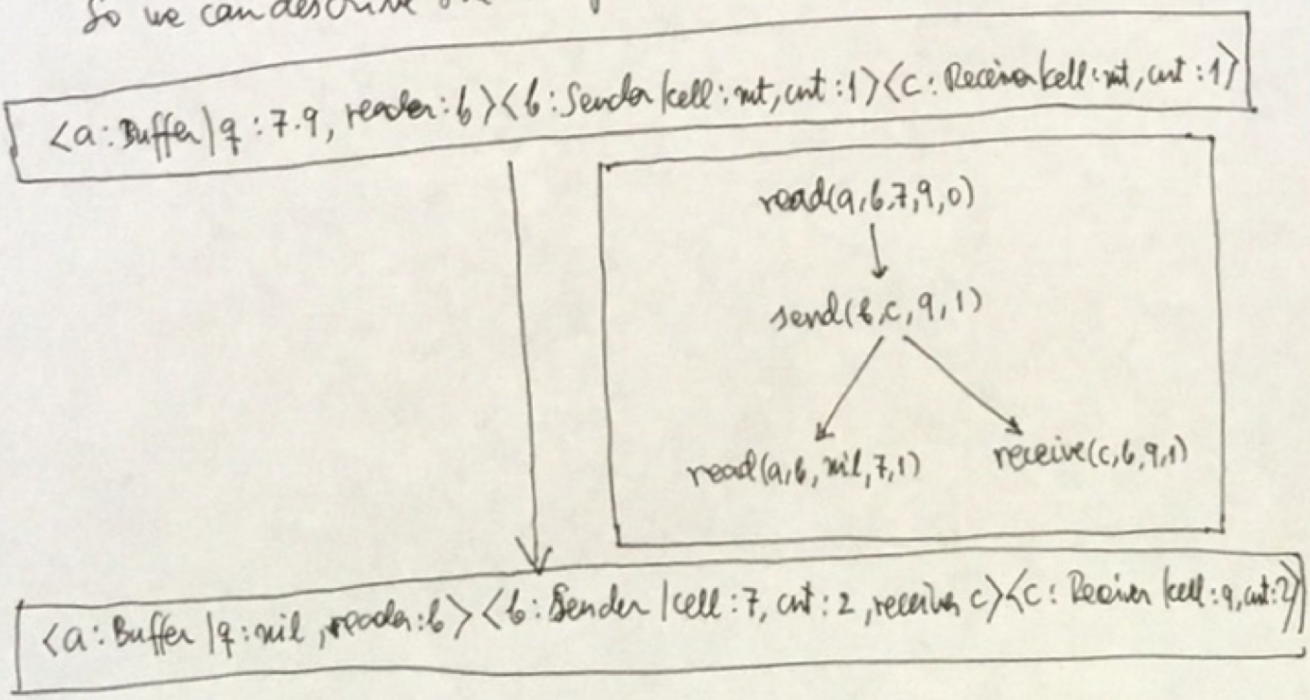
where the atomic rewrite steps have been underlined in the proof term.

Intuitively, what we want is a model of a concurrent computation as a partial order of atomic events, where atomic events are atomic rewrite steps of the form $r(u_1, .., u_n)$ where the $u_1, .., u_n \in T_\Sigma$, i.e.; there is no nested concurrency caused by additional rewrites in the $u_1, .., u_n$, since they are idle rewrites $u_1, .., u_n$. Here, the read(a, b, 7, 9, 0) must necessarily precede the send(b, c, 9, 1) event, because the sender object b must do the read before the send.

Note also that:

1. The read(a, b, nil, 1) and receive(c, b, 9, 1) are concurrent and therefore causally unrelated, but
2. they are both causally dependent on the event send(b, c, 9, 1)

So we can describe the computation as the partial order:



Coherent Configurations

Not all configurations of objects and messages make sense. A configuration of the form:

$$\langle o : C \mid atts \rangle \; \langle o : C' \mid atts' \rangle \; U$$

where o is the same object name, and C can be equal to C' or not, is incoherent: a perfect recipe for confusion.

In general, however, there can be several copies of the same message in a configuration, for example due to message resending. However, by adding a message identifier to each message, [e.g., by adding a counter to the sender object] we can also achieve message uniqueness.

Therefore, assume messages have unique id's, we can then postulate a function:

$$ids : Conf \longrightarrow M(Ids)$$

that sends its configuration to the multiset of its object and message identifiers.

Definition. A configuration C is called coherent iff $ids(C)$ is a set, i.e., there are no repeated identifiers in it.

More specifically, we will always assume that the function ids is defined recursively as follows:

$$\begin{cases} ids(null) = \emptyset \\ ids(\langle O : C | atts \rangle \, C') = O \; ids(C) \\ ids(M \, C) = ids(M) \; ids(C) \end{cases}$$

where M is a message, and $ids(M)$ is the unique identifier of that message.

The problem that we may run into is that, if the rules of an O-O system $\mathcal{R}$ have not been specified carefully, a coherent configuration C might be rewritten to an incoherent one C', i.e., one where $ids(C')$ is not a set, but a (proper) multiset.

This problem can be avoided by requiring the following property of $\mathcal{R}$:

Definition An O-O rewrite theory $\mathcal{R}$ is coherence preserving iff for any set of objects, i.e., any configuration $C_0 = \langle O_1 : C_1 | atts_1 \rangle \cdots \langle O_n : C_n | atts_n \rangle$ with $\{O_1 \cdots O_n\}$ a set and any sequence of atomic rewrites of the form $U\,t(u_1,\ldots,u_n) \xrightarrow{U\,r(u_1,\ldots,u_n)} U\,t(v_1,\ldots,v_n)$ with $r : t(x_1 \cdots x_n) \to t'(x_1 \cdots x_n) \in R$ a rule in R of the form:

$$\langle O_1 : C_1 | atts_1 \rangle \cdots \langle O_n : C_n | atts_n \rangle \xrightarrow{U_1 r(\vec{u}_1)} U_1 V_1 \cdots U_n V_n \xrightarrow{U_n r(\vec{u}_n)} U_n V_n \, V_{n+1}$$

with $n \geq 0$, we have:

(1) all configurations $U_1 V_1, \ldots, U_{n+1} V_{n+1}$ are coherent

(2) $\forall \; 1 \leq i < j \leq n+1, \quad r_i(\vec{u}_i) \neq r_j(\vec{u}_j)$ [uniqueness of atomic rewrites]

(3) Whenever $U \xrightarrow{r(u_1,\ldots,u_n)} V$, the $u_1, \ldots, u_n$ can never be rewritten.

(4) Call a configuration ~~set~~ object-set-reachable iff it is either an object set or of the form $C = U_{n+1} V_{n+1}$ in the above sequence scheme. We furthermore require that if C and C' are object-set reachable [therefore coherent by (1)] and $ids(C) \cap ids(C') \neq \emptyset$, then CC' is also object-set reachable [and therefore coherent].

For example, one can check that our Sender, Buffer and Receiver rewrite theory example $\mathcal{R}$ is coherence-preserving by defining:

$$ids\,(\text{to } Z : E \text{ from } (Y,N)) = (Y,N)$$

Defining the Partial Order of Events Model $\mathcal{E}\mathcal{R}$

Assume throughout that $\mathcal{R}$ is a coherence-preserving rewrite theory specifying a concurrent O-O systems. We can then define a subcategory $\mathcal{T}_{\mathcal{R}}^{\circ} \subseteq \mathcal{T}_{\mathcal{R}}$ whose objects are the object-set reachable configurations, and whose arrows are $C \xrightarrow{[\alpha]} C'$ in $\mathcal{T}_{\mathcal{R}}^{\circ}$ iff $C \xrightarrow{[\alpha]} C'$ in $\mathcal{T}_{\mathcal{R}}$. This is well-defined, because if C is set-object reachable, then C' is so too, thanks to the Sequentialization Lemma that states that any such $C \xrightarrow{[\alpha]} C'$ ~~can be~~ is equivalent to: $[\alpha] = [U_1 r_1(\vec{u}_1); \cdots; U_n r_n(\vec{u}_n)]$ for $n \geq 1$, or $[\alpha] = [C]$ and $C = C'$. [note that C, C' are EVACV-equivalent classes].

Note that $\mathcal{T}^\circ_{\mathcal{R}}$ is not closed in general under the multitot union operation $\left(U_1 \xrightarrow{[\alpha_1]} V_1,\ U_2 \xrightarrow{[\alpha_2]} V_2\right) \mapsto U_1 U_2 \xrightarrow{[\alpha_1 \alpha_2]} V_1 V_2$ because, for example, $U_1 U_1$ is not a coherent configuration unless $U_1 = \text{null}$. However, it is closed under such union operation iff $ids(U_1) \cap ids(U_2) = \emptyset$, i.e., if these are disjoint object-set-reachable configurations.

The category $\mathcal{E}_{\mathcal{R}}$ of Computations as Partial Orders of Events

Define the set $\underline{Events}_{\mathcal{R}}$ of the [atomic] events in $\mathcal{R}$ as:

$$\underline{Events}_{\mathcal{R}} = \{[r(u_1, \ldots, u_n)] \mid u_1, \ldots, u_n \in T_\Sigma \wedge r(u_1, \ldots, u_n) \in Proof\text{-}Terms_{\mathcal{R}}\}$$

We will associate to each computation $U \xrightarrow{[\alpha]} V$ in $\mathcal{T}^\circ_{\mathcal{R}}$ a partial order of events computation $U \xrightarrow{\langle Ex, <_\alpha \rangle} U$, where $Ex \subseteq \underline{Events}_{\mathcal{R}}$ is a finite set of events in $\mathcal{R}$, and $<_\alpha$ is a strict partial order on those events. We can do so inductively by associating to each proof term α, where $[\alpha]$ labels an arrow in $\mathcal{T}^\circ_{\mathcal{R}}$ a partially ordered set ~~as follows: done~~ denoted $[\![\alpha]\!]$ as follows:

1. $[\![\bar{U}]\!] = \langle \emptyset, \emptyset \rangle$ $\bar{U}$ an object readable configuration

2. $[\![r(u_1, .., u_n)]\!] = \langle \{ r(u_1, .., u_n) \}, \emptyset \rangle$

where $u_1, .., u_n \in T_\Sigma$

3. $[\![\alpha \; \beta]\!] = \langle \mathcal{E}_\alpha, <_\alpha \rangle \cup \langle \mathcal{E}_\beta, <_\beta \rangle$

4. $[\![\alpha ; \beta]\!] = \langle \mathcal{E}_\alpha \cup \mathcal{E}_\beta, <_{\alpha;\beta} \rangle$, where

$$<_{\alpha;\beta} = \left(<_\alpha \cup <_\beta \cup \{ e_0 < e_1 \mid e_0 \in \mathcal{E}_\alpha, e_1 \in \mathcal{E}_\alpha, ids(e_0) \cap ids(e_1) \neq \emptyset \} \right)^+$$

where $(\_)^+$ denotes transitive closure, and for any event $U \xrightarrow{r(u_1, .., u_n)} V$, by definition $ids(r(u_1, .., u_n)) = ids(\bar{U}) \cup ids(V)$.

A few remarks are in order to explain why the map $[\alpha] \mapsto \langle \mathcal{E}_\alpha, <_\alpha \rangle$ is well-defined, i.e., it does not depend on the choice of $\alpha' \in [\alpha]$ and defines a partial order. The two reasons are, essentially, the facts that: (i) $\mathcal{R}$ is coherence-preserving, and (ii) the Sequentialization lemma. They ensure that:

- in (2) all the posets have disjoint sets of elements, so their union is a poset
- in (3) the same is the case, since α and β are disjoint compu-

tations

- by the Sequentialization Lemma one can show that if $\alpha_1, \alpha_2 \in [\alpha]$, then $[\![\alpha_1]\!] = [\![\alpha_2]\!]$.
- by condition (3) of $\mathcal{R}$ coherence-preserv, any $r(\alpha_1 \cdots \alpha_n)$ is of the form $r(u_1 \cdots u_n)$

Remark. In the definition of $\mathcal{T}_{\mathcal{R}}^{\circ}$, and therefore of $\mathcal{E}_{\mathcal{R}}$, one could question whether the requirement that a configuration C is object-set-reachable is not too restrictive. But there is no real loss of generality. We can always simulate the presence of some messages in an initial configuration by assuming a class Start of start objects, only used at the beginning, with rules of the form:

$$r: \langle o : \text{Start} \mid \text{msg} : \text{mexp} \rangle \longrightarrow \text{mexp}$$

where $\text{mexp} = \text{mexp}(x_1, \ldots, x_n)$ is a message expression. That is, a Start object just introduces a message in the configuration and garbage-collects itself.

We can now define the category $\mathcal{E}_{\mathcal{R}}$ of partial order computations of $\mathcal{R}$ with:

- U object iff U is a object-set-reachable configuration
- $U \xrightarrow{\langle \mathcal{E}, < \rangle} V$ arrow iff $\langle \mathcal{E}, < \rangle = \langle \mathcal{E}_\alpha, <_\alpha \rangle$ for some $U \xrightarrow{[\alpha]} V$ in $\mathcal{T}_{\mathcal{R}}^{\circ}$

Main Theorem. The labeled graph homomorphism

$$[\![-]\!] : \mathcal{T}_R^\circ \to \mathcal{E}_R^\circ : (U \xrightarrow{[\alpha]} V) \mapsto (U \xrightarrow{\langle \mathcal{E}_\alpha, \prec_\alpha \rangle} V)$$

(a functor and) is an isomorphism of categories, and preserves the disjoint union of computations operations in both categories.

Proof Sketch. For the theorem to make sense, we of course should define explicitly arrow composition in $\mathcal{E}_R$ as follows: Given $U \xrightarrow{\langle \mathcal{E}_1, \prec_1 \rangle} V \xrightarrow{\langle \mathcal{E}_2, \prec_2 \rangle} W$, then

$$\langle \mathcal{E}_1, \prec_1 \rangle ; \langle \mathcal{E}_2, \prec_2 \rangle = \langle \mathcal{E}_1 \cup \mathcal{E}_2, \prec_1 ; \prec_2 \rangle,$$

where $\prec_1 ; \prec_2$ is defined exactly as $\prec_{\alpha;\beta}$ in pg. 8.

Functoriality is then obvious. A functor $F : \mathcal{A} \to \mathcal{B}$ between two categories is called an isomorphism of categories iff there is another functor $G : \mathcal{B} \to \mathcal{A}$ such that (i) $F;G = 1_{\mathcal{A}}$ and (ii) $G;F = 1_{\mathcal{B}}$. This means that $\mathcal{A}$ and $\mathcal{B}$ are equivalent representations of the same concept.

For the rest of the proof see the Meseguer-Talcott paper.