CS 524
Lecture 25
J. Meseguer

1. The Canonical Transition System $\mathcal{C}_{\mathcal{R}}$ Associated to a Ground Coherent Theory $\mathcal{R}$

We have just seen that, if $\mathcal{R} = (\Sigma, \phi, E \cup B, R)$ is a ground coherent rewrite theory, any <u>one-step transition</u> $[u]_{E \cup B} \xrightarrow{\mathcal{R}} [v]_{E \cup B}$ in $\mathcal{R}$ has a corresponding one-step transition:

$[u\,!_{E/B}]_B \xrightarrow{\mathcal{R}} [v\,!_{E/B}]$ given by the equivalence:

$$[u\,!_{E/B}]_B \xrightarrow{\mathcal{R}} [v\,!_{E/B}] \Leftrightarrow u \xrightarrow[R^{\phi}/B;\, E/B!]{} v' =_B v\,!_{E/B}$$

and, of course, one-step rewrite transitions in $\mathcal{R}$ define a <u>transition system</u> $(T_{\Sigma/E \cup B}, \xrightarrow{\mathcal{R}})$ whose <u>states</u> are the elements of the algebraic data type [initial algebra] defined by the equational theory $(\Sigma, E \cup B)$.

This raises an obvious question: how can we best describe the transition system at the level of the <u>unique representatives</u> $[u\,!_{E/B}]_B$ of each equivalence class $[u]_{E \cup B} \in T_{\Sigma/E \cup B}$?

An obvious sub-question is the following: $T_{\Sigma/E \cup B}$ is not just a <u>set</u> of states, but an [initial] Σ-algebra. Can we likewise describe the set of states in normal form as an <u>isomorphic</u> [and therefore also initial] Σ-algebra?

The answer to this sub-question is a rotound <u>Yes</u>!

Definition. Let $(\Sigma, E \cup B)$ be a convergent equational theory. The *canonical term algebra* defined by $(\Sigma, E \cup B)$, $C_{\Sigma/E,B}$ has, for each sort $s \in S$ in Σ a set of elements:

$$C_{\Sigma/E,B,s} = \{ [u]_B \in T_{\Sigma/B} \mid [u]_B = [u!_{E/B}]_B \wedge u \in T_{\Sigma,s} \}$$

Here we are assuming that Σ is what is call B-*preregular* [a property automatically checked by Maude], i.e;

$$(\forall s \in S) \quad u \in T_{\Sigma,s} \wedge u' =_B u \Rightarrow u' \in T_{\Sigma,s}$$

so that the [least] sort of a B-equivalence class $[u]_B$ can be determined by computing that of *any* representative $u' \in [u]_B$

$C_{\Sigma/E \cup B}$ as a natural Σ-*algebra* structure given as follows:

1. For any *constant* $a : \to s$, its interpretation in $C_{\Sigma/E \cup B}$ is the B-equivalence class: $[a!_{E/B}]_B \in C_{\Sigma/E \cup B}$.

2. For any $f : s_1 \dots s_n \to s$ in Σ and any $([u_1], \dots, [u_n]) \in C_{\Sigma/E,B,s_1} \times \cdots \times C_{\Sigma/E,B,s_n}$, the interpretation of f is the *function* $f_{C/E,B}$ mapping $([u_1], \dots, [u_n])$ to the B-equivalence class: $[f(u_1, \dots u_n)!_{E/B}] \in C_{\Sigma/E,B,s}$

Note that $C_{\Sigma/E,B}$ has a *very concrete meaning*: it is the algebra [implicitly] defined by Maude's *reduce* command, since the result of *reduce* t . is precisely $t!_{E/B}$.

For example, for the Maude functional module

```
fmod NATURAL is
sort Nat.
op 0 : -> Nat [ctor].
op s : Nat -> Nat [ctor].
op _+_ : Nat Nat -> Nat.
vas N M : Nat
eq N+0 = N.
eq N+s(M) = s(N+M)
endfm
```

The canonical term algebra $C_{\Sigma/E}$ has as its elements the set $\mathbb{N} = \{ s^n(0) \mid n \in \mathbb{N} \}$, where $s^0(0) = 0$ by convention, i.e., the set of natural numbers in Peano notation, and the function

$$(-+-)_{C/\Sigma/E} : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$
$$(s^n(0), s^m(0)) \longmapsto s^{n+m}(0)$$

is precisely the addition function on natural numbers, which is what the reduce $s^n(0) + s^m(0)$. command computes in Maude for this module.

Theorem. For any [ground] convergent equational theory $(\Sigma, E \cup B)$ the function:

$$!_{E/B} : T_{\Sigma/E \cup B} \ni [u]_{E \cup B} \longmapsto [u!_{E/B}] \in C_{\Sigma/E,B}$$

defines an isomorphism of Σ-algebras $T_{\Sigma/E \cup B} \cong C_{\Sigma/E,B}$

Proof. Left as an exercise. Hint: use the Church-Rosser Equivalence: $u =_{E \cup B} v \iff u!_{E/B} =_B v!_{E/B}$.

So, at last we can define our desired one-step transition system for a ground coherent rewrite theory $\mathcal{R} = (\Sigma, \emptyset, E \cup B, R)$:

$$\mathcal{G}_{\mathcal{R}} =_{def} (C_{\Sigma/E,B}, \to_{\mathcal{R}})$$

Furthermore, the isomorphism: $!_{E/B} : T_{\Sigma/E \cup B} \xrightarrow{\simeq} C_{\Sigma/E,B}$ defines, likewise an isomorphism of transition systems

$$!_{E/B} : (T_{\Sigma/E \cup B}, \to_{\mathcal{R}}) \xrightarrow{\simeq} \mathcal{G}_{\mathcal{R}}$$

$\mathcal{G}_{\mathcal{R}}$ is of course essential in Maude. For example:

1. Any rewrite sequence exhibited [by setting trace on] associated to n+1 steps of the command rewrite t. has the form:

$$[t!_{E/B}]_B \to_{\mathcal{R}} [u_1] \to_{\mathcal{R}} [u_2] \ldots \to [u_n] \to [u_{n+1}]$$

with $[u_1], \ldots, [u_n] \in C_{\Sigma/E,B}$

2. The search graph generated by Maude's search $t \Rightarrow^* u(x_1, \ldots, x_n)$. is a graph rooted at $[t!_{E/B}]$ whose elements belong to $C_{\Sigma/E,B}$ and whose edges are transitions $[u] \to_{\mathcal{R}} [v]$ in $\mathcal{G}_{\mathcal{R}}$

3. The same happens when we give commands to Maude's LTL model checker: its search space is a graph for $\mathcal{G}_{\mathcal{R}}$.

The great thing is that, thanks to:

1. The isomorphism $(T_{\Sigma/E \cup B}, \to_R) \cong \mathcal{C}_R$, and
2. The Sequentialization Theorem for any category of computation $\mathcal{T}_R$ associated to any rewrite theory R

we exactly capture all and do not miss any computation $[u] \xrightarrow{[\alpha]} [v]$ in $\mathcal{T}_R$: Maude's anaylis is sound and complete for $\mathcal{T}_R$, provided R is ground coherent.

2. End Game: The Big Picture

What have we done this semester in CS 524?

1. Studied a quite extensive family of models of concurrency
2. Shown that any such model of concurrency can be naturally [i.e., without any ugly encoding, with 0 or ε representational distance] be formally specified by a rewrite theory R so that:
   a. its concurrent computations can be characterized as deductions in the logic $\mathcal{L}(R)$ associated to R
   b. Such concurrent computations define a category $\mathcal{T}_R$ that provides a true concurrency model of the concurrent system so specified

3. Seen that this approach gives rise to the rewriting logic semantics project for concurrent programming languages, which has been advanced jointly with Prof. Grigore Roşu at UIUC using both Maude and ℍ

4. Characterized important classes of concurrent systems such as:
   a. Concurrent Object Systems
   b. Process Calculi
   c. Lambda Calculi and Combinatory Logic
   d. Deterministic Systems
   e. Sequential Systems
   f. Petri Nets
   g. Grammars

5. Seen how the property that the reachable states from some initial state init satisfy an invariant predicate Inv can be checked by giving the Maude command:

   search init $\Rightarrow^*$ X:State such that Inv(X) =/= true.

   provided $\mathcal{R}$ is ground coherent, and

6. Studied the ground coherence property of a rewrite theory $\mathcal{R}$ as the key executability condition [assuming ground coherence of $(\Sigma, E \cup B)$] to execute rewrite theories in Maude

3. Where to go from here

If we had had more time, the next natural step would have been to study how temporal logic properties, both liveness and safety ones, of a concurrent system specified by a rewrite theory $\mathcal{R}$ can be verified in Maude by:

a. The LTL Model checker provided by Maude

b. The LTLR Model checker that can verify richer properties, including verification under fairness conditions, and is available on the Maude web page

c. The symbolic, logical LTL model checker, which can start from symbolic initial states (terms with variables describing a typically infinite set of initial states as instances) and may be able to verify LTL properties for infinite sets of states reachable from the [already infinite] symbolic initial state, using the tool also available in the Maude web page.

We also would have been able to discuss how rewrite theories can be extended to:

a. Real-time rewrite theories modeling concurrent real-time systems, and supported by the Real-Time Maude tool and its model checker for verification purposes, and

b. Probabilistic rewrite theories, analyzable in their quantitative and qualitative properties by statistical model checking.

Furthermore, we would have studied reachability logic and its Maude-based theorem prover, to verify reachability properties [including Hoare logic properties and invariants] of rewrite theories [see paper in the ~~Maude~~ course web page].

Finally, we could have surveyed many applications of rewriting logic, of which some of the term projects and the paper presentations have given you some samples.

In the area of applications I would suggest reading two papers, both in the course web page:

a. 20 Years of Rewriting Logic

b. The survey paper on Cloud Transaction Systems Applications [since this whole area was developed after the 20 Years paper was published and is not discussed there].