Term Rewriting Systems — first look

Tern rewriting systems are used to obtain an operational semantics for equational specifications. Each equation is oriented from left-to-right and is used in a "substitute equals for equals" scheme.

Definition: a **substitution** σ is a mapping from variables to terms, σ: vars $\rightarrow$ terms. A **substitution is applied** a term by replacing each occurrence of a variable V by $\sigma(V)$. If V is not mapped by σ, assume $\sigma(V) = V$.

For example if $\sigma(X) = X + 1$, then applying σ to X + Y yields (X + 1) + Y.

Equations as rewrite rules:
- equation t1 = t2 becomes rewrite rule t1 $\Rightarrow$ t2,
- a rewrite rule t1 $\Rightarrow$ t2 is applied to a term t by finding a subterm t' of t so that σ(t1) = t', and then replacing t' by σ(t2) in t. We write t1 $\Rightarrow^*$ t2 provided that for some t3, t4, … t1 $\Rightarrow$ t3 $\Rightarrow$ … $\Rightarrow$ t4 $\Rightarrow$ t2.

Example
0 + X = X
s X + Y = s(X + Y)
0 * X = 0
(s X) * Y = (X * Y) + X

Then
0 + (s 0 * 0)
$\Rightarrow$ s 0 * 0 by rule 1
$\Rightarrow$ (0 * 0) + 0 by rule 4
$\Rightarrow$ 0 + 0 by rule 3
$\Rightarrow$ 0 by rule 1

Definition: a collection of rewrite rules is called **terminating** if there is no infinite chain t1 $\Rightarrow$ t2 $\Rightarrow$ t3 $\Rightarrow$ …

The rewriting rules illustrated above are terminating, but in general this is not easy to determine. There are very simple rules that are not terminating (e.g., X + Y = Y + X), and when several rules cooperate to cause non-termination it is genuinely difficult to detect.

Definition: a term that cannot be rewritten as another is said to be in **normal form**; if we start from a term and perform rewriting until it is no longer possible, the result is referred to as the *normal form of the original*. For instance, for the

system above, s(s 0) + s 0 $\overset{2}{\Rightarrow}$ s(s 0 +s 0) $\overset{2}{\Rightarrow}$ s(s(0 + s 0)) $\overset{1}{\Rightarrow}$ s(s(s 0)), and this last term is the normal form of s(s 0) + s 0.

Definition: a collection of rewrite rules is called **confluent** if whenever t1 ⇒ t2 and t1 ⇒ t3, then there is t4 so that t2 ⇒* t4 and t3 ⇒* t4.

The options for rewriting a term with different rules will often be numerous, and may even be so using only one rule. For instance, with the above system, we have s(0 + s 0) + s 0 $\overset{1}{\Rightarrow}$ s(s 0) + s 0 and s(0 + s 0) + s 0 $\overset{2}{\Rightarrow}$ s(0 + s 0 + s 0). The confluence property guaranties that the choice among such alternatives is irrelevant in the sense that they will all converge to a common result by further rewriting. The system above is also confluent, but again this may not be easy to determine. In the instance above, s(s 0) + s 0 $\overset{2}{\Rightarrow}$ s(s 0 + s 0), and s(0 + s 0 + s 0) $\overset{1}{\Rightarrow}$ s(s 0 + s 0).

Lemma: if a rewriting system R is confluent, then the normal form of any term, if it exists, is unique.

Definition: if R is a terminating rewriting system and ≡ is its corresponding equivalence relation (i.e., view each ⇒ as =), then R is **canonical** if for every pair of terms t_1 and t_2, $t_1 \equiv t_2$ if and only if t_1 and t_2 have the same normal form. Hence, every term of each equivalence class rewrites to a unique normal form, the *canonical form of the class*.

Theorem: a rewriting system is canonical if and only if it is terminating and confluent.

According to this theorem, the equational axiom systems where this rewriting animation is faithful to equational logic are those that are terminating and confluent. Systems where one or both of these properties fail will produce misleading results when animated in this way.

Note that for ADTs, we are interested in applying rewrite rules to ground terms as an aid to determining the equivalence classes of the initial algebra. In fact, CafeOBJ formally restricts the application of operations such as 'reduce' to ground terms (but often doesn’t enforce it).