AR TUTORIAL 6 - Rewrite Rules and Induction

EXERCISE ONE: Critical Pairs

Find a non-trivial critical pair of the following rewrite rules, where x, y, u, v, w are variables.

$$s(x)+y \Rightarrow x+s(y) \text{ and } (u+v)+w \Rightarrow u+(v+w)$$

Explain your reasoning.

mgu $s(x) = (u+v)$ $\quad$ $x=u \rightarrow s(u) = u+w$ $\quad$ $s(x) = x+v$

$\quad s(y) = (v+w)$ $\quad$ $y=w \rightarrow s(w) = v+w$ $\quad$ $s(y) = v+y$

Therefore, $s(\circ) = (\circ + v)$

RULE: $\langle R_1\theta, L_1\theta\{R_2\theta / s'\theta\}\rangle$

$\theta = [s(x)/u, y/v]$

$R_1\theta = s(x) + (y+w)$

$L_1\theta\{R_2\theta/s'\theta\} = (x+s(y))+w$

SOLUTION

Recall from lectures that for rewrite rules $L_1 \Rightarrow R_1$ and $L_2 \Rightarrow R_2$, a critical pair can be defined as:

$$\langle R_1[\theta], L_1[\theta]/\{R_2[\theta]/s\}\rangle$$

where θ = mgu of s (subpart of L_1) and L_2.

Now if we take:

$$\overbrace{\underbrace{(u+v)}_{s}+w}^{L_1} \Rightarrow \overbrace{u+(v+w)}^{R_1}$$

$$\overbrace{s(x)+y}^{L_2} \Rightarrow \overbrace{x+s(y)}^{R_2}$$

Then $\theta = [s(x)/u, y/v]$, so the critical pair is given by

$$\langle s(x)+(y+w), (x+s(y))+w\rangle$$

More concisely: The expression $s(x)+y$ unifies with $u+v$ with common instance $s(x)+y$. $(s(x)+y)+w$ can be rewritten to either $(x+s(y))+w$ or $s(x)+(y+w)$.

So, the critical pair is:

$$\langle s(x)+(y+w), (x+s(y))+w\rangle$$

EXERCISE TWO : Confluence

Consider the following rewrite rules, where X, Y, and Z are variables:

$$(X \cdot Y) \cdot Z \Rightarrow X \cdot (Y \cdot Z)$$

$$X \cdot 0 \Rightarrow X \qquad Q \cdot 0 \Rightarrow Q$$

$$X \cdot i(X) \Rightarrow 0 \qquad W \cdot i(W) \Rightarrow 0$$

Give two ways in which the system of rewrite rules is not locally confluent. Explain in terms of critical pairs.

* An element $a \in S$ is said to be locally confluent if for all $b, c \in S$ with $a \to b$ and $a \to c$, there exists $d \in S$ with $b^* \to d$ and $c^* \to d$.

$\{Q \cdot Z; Q(0 \cdot Z)\}$

$\{0 \cdot Z; W \cdot (i(W) \cdot Z)\}$

$X \cdot (Y \cdot Z)$

→ can't rewrite $Q \cdot Z$ to $Q \cdot 0$: you get stuck → not locally confluent!

SOLUTION

Two examples of critical pairs that are not joinable are:

1. Starting from $(X \cdot 0) \cdot Z$ we can get the critical pair $\langle X \cdot (0 \cdot Z), X \cdot Z \rangle$
2. Starting from $(X \cdot i(X)) \cdot Z$ we can get the critical pair $\langle 0 \cdot Z, X \cdot (i(X) \cdot Z) \rangle$

The critical pair is not joinable (or 'conflatable').

EXERCISE THREE : Induction

Consider the following (Isabelle) datatype definition:

datatype 'a TREE = LEAF 'a | NODE 'a "'a TREE" "'a TREE"

1. Give an induction type rule appropriate for proofs by (structural) induction involving the TREE datatype.

$$\left[\begin{array}{l} \phi(\text{Leaf}) \wedge \\ \forall v, t_1, t_2 \cdot [\phi(t_1) \wedge \phi(t_2)) \to \phi(\text{Node}(v, t_1, t_2)) \end{array}\right] \to \forall t \cdot \phi t$$

alternative: $\dfrac{\Gamma \vdash P(\text{LEAF}\, x) \quad \Gamma, P(x_1), P(x_2) \vdash P(\text{NODE}\; a\; x_1\; x_2)}{\Gamma \vdash \forall t \cdot P(t)}$

2. Define a function MIRROR that recursively flips the nodes in the left and right subtrees of a tree as defined above.

ALGORITHM

1. MIRROR (t_1) → left subtree
2. MIRROR (t_2) → right subtree
3. Swap left and right subtrees

MIRROR (LEAF x) = LEAF x
MIRROR (NODE x LEFT RIGHT) =
NODE x [MIRROR RIGHT] [MIRROR LEFT]

* use primrec MIRROR:: "'a TREE ⇒ 'a TREE"

SOLUTION

```
primrec MIRROR:: "'a TREE ⇒ 'a TREE" where
  "MIRROR (LEAF x) = LEAF x"
| "MIRROR (NODE x l r) = NODE x (MIRROR r) (MIRROR l)"
```

3. Formalize your definition of MIRROR in Isabelle and give a structured (Isar) proof that MIRROR (MIRROR t) = t.

```
lemma "MIRROR (MIRROR t) = t"
proof (induction t)
    case LEAF
    then show ?case by simp
next
    case (NODE x left right)
    then show ?case by simp
qed
```