

Given these sentences, a standard inference procedure such as resolution can perform tasks requiring equality reasoning, such as solving mathematical equations. However, these axioms will generate a lot of conclusions, most of them not helpful to a proof. So the second approach is to add inference rules rather than axioms. The simplest rule, **demodulation**, takes a unit clause  $x=y$  and some clause  $\alpha$  that contains the term  $x$ , and yields a new clause formed by substituting  $y$  for  $x$  within  $\alpha$ . It works if the term within  $\alpha$  unifies with  $x$ ; it need not be exactly equal to  $x$ . Note that demodulation is directional; given  $x=y$ , the  $x$  always gets replaced with  $y$ , never vice versa. That means that demodulation can be used for simplifying expressions using demodulators such as  $z+0=z$  or  $z^1=z$ . As another example, given

$$\begin{aligned} & \text{Father}(\text{Father}(x)) = \text{PaternalGrandfather}(x) \\ & \text{Birthdate}(\text{Father}(\text{Father}(\text{Bella})), 1926) \end{aligned}$$

we can conclude by demodulation

$$\text{Birthdate}(\text{PaternalGrandfather}(\text{Bella}), 1926).$$

More formally, we have

- **Demodulation:** For any terms  $x$ ,  $y$ , and  $z$ , where  $z$  appears somewhere in literal  $m_i$  and where  $\text{UNIFY}(x, z) = \theta \neq \text{failure}$ , Demodulation

$$\frac{x=y, \quad m_1 \vee \cdots \vee m_n}{\text{SUB}(\text{SUBST}(\theta, x), \text{SUBST}(\theta, y), m_1 \vee \cdots \vee m_n)}.$$

where  $\text{SUBST}$  is the usual substitution of a binding list, and  $\text{SUB}(x, y, m)$  means to replace  $x$  with  $y$  somewhere within  $m$ .

The rule can also be extended to handle non-unit clauses in which an equal sign appears:

- **Paramodulation:** For any terms  $x$ ,  $y$ , and  $z$ , where  $z$  appears somewhere in literal  $m_i$ , and where  $\text{UNIFY}(x, z) = \theta \neq \text{failure}$ , Paramodulation

$$\frac{\ell_1 \vee \cdots \vee \ell_k \vee x=y, \quad m_1 \vee \cdots \vee m_n}{\text{SUB}(\text{SUBST}(\theta, x), \text{SUBST}(\theta, y), \text{SUBST}(\theta, \ell_1 \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_n))}.$$

For example, from

$$P(F(x, B), x) \vee Q(x) \quad \text{and} \quad F(A, y) = y \vee R(y)$$

we have  $\theta = \text{UNIFY}(F(A, y), F(x, B)) = \{x/A, y/B\}$ , and we can conclude by paramodulation the sentence

$$P(B, A) \vee Q(A) \vee R(B).$$

Paramodulation yields a complete inference procedure for first-order logic with equality.

A third approach handles equality reasoning entirely within an extended unification algorithm. That is, terms are unifiable if they are *provably* equal under some substitution, where “provably” allows for equality reasoning. For example, the terms  $1+2$  and  $2+1$  normally are not unifiable, but a unification algorithm that knows that  $x+y=y+x$  could unify them with the empty substitution. **Equational unification** of this kind can be done with efficient algorithms designed for the particular axioms used (commutativity, associativity, and so on) rather than through explicit inference with those axioms. Theorem provers using this technique are closely related to the CLP systems described in Section 9.4. Equational unification