Chris Carpio
05carpio
994954518

Cannot be expressed in Prolog:
1a, 1c and 1d cannot be expressed in Prolog.

1a cannot be expressed in Prolog within the specification because of the limitations due to the special operator rule. The explicit ban of ! (Cut) or the equivalent special operator of Not hinders the ability to check if the person is actually the opposite sex. Negation by failure plays a heavy role in this.

1c cannot be expressed because Prolog cannot express “some” in a cohesive way. It would either be that they have children, or they do not have children.

1d can indeed be expressed in Prolog but not without ambiguity. Depending on the order of code Prolog can conclude that they can indeed be one type of sibling, but never the other due to the iterative nature of Prolog.

First-order logic:

1. ∀x ~(male(x) ^ female(x))
2. ∀x ∀y (female(y) ^ parent(y, x))
3. ∀x ∀y ∃z ~(parent(x, z) ^ parent(y, z))
4. ∃x ∃y ((male(x) ^ male(y)) v (female(x) ^ female(y)))
-> brothers(x, y) v sisters(x, y)
5. ∃x ∃y ∃z (parent(z, x) v parent(z, y)) -> siblings(x, y)
6. ∃x ∃y ∃a ∃b ((parent(a, x) v parent(b, y)) ^ siblings(a, b)) -> cousins(x, y)
7. ∃x ∃y ∃a (cousin(x, y) ^ cousin(y, a)) -> cousin(x, a)