Lecture 25

Pete Manolios
Northeastern

Connections with ACL2

For any FO ϕ, we can find a universal ψ in an *expanded* language such that ϕ is satisfiable iff ψ is satisfiable.

$\langle \forall u, v \ \langle \exists z \ \phi(u, v, z) \rangle\rangle$ $\langle \forall u, v \ \langle \exists z \ (App \ u \ v) = (Rev \ z) \rangle\rangle$

First, PNF, and push existentials left (2nd order logic)

$\langle \exists F_z \ \langle \forall u, v \ \phi(u, v, F_z(u, v)) \rangle\rangle$ $\langle \exists F_z \ \langle \forall u, v \ (App \ u \ v) = (Rev \ (F_z \ u \ v)) \rangle\rangle$

Previously, we saw how to go back to FO while preserving SAT with

$\langle \forall u, v \ \phi(u, v, F_z(u, v)) \rangle$ $\langle \forall u, v \ (App \ u \ v) = (Rev \ (F_z \ u \ v)) \rangle$

But what about preserving validity? This method doesn't work, as we've seen. Can we make it work in a FO setting?

This is how ACL2 handles quantifiers

DEMO

$$\langle \forall u, v \ \langle \exists z \ (App \ u \ v) = (Rev \ z) \rangle\rangle$$

$$\longrightarrow$$

$$\langle \forall u, v \ (E_z \ u \ v) \rangle$$

As above, but not enough

$$(E_z \ u \ v) \ \equiv \ (App \ u \ v) = (Rev \ (F_z \ u \ v))$$

Constrain F_z:

$$(App \ u \ v) = (Rev \ z) \ \Rightarrow \ (E_z \ u \ v)$$

if (App *u v*) = (Rev *z*) has solution then F_z is also a solution