Program Verification - Message system

Beerend Lauwers and Frank Wijmans

February 27, 2012

Privacy safety property

Safety property
safe
(f: $(num \rightarrow num \rightarrow num)$) $(dataInp : (num\#num\#num)set) =$
$\forall$y orig cons. (y, orig, cons) IN dataInp $\Rightarrow$
((cons=0) $\Rightarrow$ (y=orig)) $\wedge$
((cons=1) $\Rightarrow$ ((y=orig) $\vee$ (f y orig))) $\wedge$
((cons=2) $\Rightarrow$ ((y=orig) $\vee$ (f y orig) $\vee$ ($\exists$x. f y x $\vee$ f x orig))) $\wedge$
((cons=3) $\Rightarrow$ ($\top$))

val safetyAlgFol = prove (–‘forward algFol 1 2 2‘– ,
(REWRITE_TAC [safe_def, network_def, data_def, follower_def,
forward_def, algFol_def, isInNetwork_def]) THEN (RW_TAC
(std_ss++PRED_SET_ss) []));

Algorithms

```
val algFol_def = Define ‘algFol d n c t f =
if (c = 0)
then (if (t=f)
then ((t,f,c) INSERT d)
else d )
else ( if (c = 1)
then (if (isFollower f t)
then ((t,f,c) INSERT d)
else (d) )
else
if (c = 2)
then (if (isFollower f t)
then ((t,f,c) INSERT d)
else ( if (∃y. (isFollower y t ∧ isFollower f y))
then ((t,f,c) INSERT d)
else d ))
else (if (c = 3)
then (if (isInNetwork t)
then ((t,f,c) INSERT d)
else(d))
else(d)))‘;
```