CNF-3SAT Reduction

Jatin Shah

October 17, 2003

$$\frac{\Gamma, x \vdash F \Rightarrow F'}{\Gamma \vdash \textsf{new}\ x.F \Rightarrow \textsf{new}\ x.F'}\ \mathit{convpar}$$

$$\frac{\Gamma \vdash F\ \textsf{literal}}{\Gamma \vdash F \Rightarrow \textsf{new}\ a\ b.(F \vee a \vee b) \wedge (F \vee a \vee \bar{b}) \wedge (F \vee \bar{a} \vee b) \wedge (F \vee \bar{a} \vee \bar{b})}\ \mathit{conv}_1$$

$$\frac{\Gamma \vdash F_1\ \textsf{literal} \quad \Gamma \vdash F_2\ \textsf{literal}}{\Gamma \vdash (F_1 \vee F_2) \Rightarrow \textsf{new}\ a.(F_1 \vee F_2 \vee a) \wedge (F_1 \vee F_2 \vee \bar{a})}\ \mathit{conv}_2$$

$$\frac{\Gamma \vdash F_1\ \textsf{literal} \quad \Gamma \vdash F_2\ \textsf{literal} \quad \Gamma \vdash F_3\ \textsf{literal}}{\Gamma \vdash (F_1 \vee F_2 \vee F_3) \Rightarrow (F_1 \vee F_2 \vee F_3)}\ \mathit{conv}_3$$

$$\frac{\Gamma \vdash F_1\ \textsf{literal} \quad \Gamma \vdash F_2\ \textsf{literal} \quad \Gamma \vdash F_3\ \textsf{literal} \quad \Gamma, a \vdash (\bar{a} \vee F_3 \vee F) \Rightarrow F'}{\Gamma \vdash (F_1 \vee F_2 \vee F_3 \vee F) \Rightarrow \textsf{new}\ a.(a \vee F_1 \vee F_2) \wedge F'}\ \mathit{conv4}$$

In Twelf:

```
% Conversion of CNF to 3CNF
conv : o -> o -> type.                          %name conv CV.
convpar : conv (new F) (new F')
           <- ({v:v} conv (F v) (F' v)).
conv1 : conv F
                (new [a] new [b] (F \/ (pos a) \/ (pos b))/\
                                 (F \/ (pos a) \/ (neg b))/\
                                 (F \/ (neg a) \/ (pos b))/\
                                 (F \/ (neg a) \/ (neg b)))
                 <- literal F.
conv2 : conv (F1 \/ F2)
             (new [a] ((pos a) \/ F1 \/ F2)/\ ((neg a) \/ F1 \/ F2))
        <- literal F1
        <- literal F2.
conv3 : conv (F1 \/ F2 \/ F3) (F1 \/ F2 \/ F3)
         <- literal F1
         <- literal F2
         <- literal F3.
conv4 : conv (F1 \/ F2 \/ F3 \/ F) (new [a] ((pos a) \/ F1 \/ F2) /\ (F' a))
         <- literal F1
         <- literal F2
         <- literal F3
         <- ({v:v} conv (neg v \/ F3 \/ F) (F' v)).
conv/\ : conv (F1 /\ F2) (F1' /\ F2')
          <- conv F1 F1'
          <- conv F2 F2'.
```

For example, if `F=new ([x:v] new ([y:v] pos y \/ neg x))` and `ALG: conv F F'` for some `F':o`, then `ALG=convpar ([v1:v] convpar ([v2:v]  conv2 lneg lpos))`.

We have to figure out if there is natural way to define *locality* for these terms.