Peter Neubert, Leona Schlöder July 7, 2014

Hauptseminar mathematische Logik
Herbrandmodell und Unifikation

Aufgabe 1. Bestimme Herbranduniversum von einer gegebenen Formel:

$$
\begin{aligned}
\Phi_1 &= \exists x \quad \phi(x) \\
\Phi_2 &= \forall x (x = 0 \vee \exists y \quad x = y + 1) \\
\Phi_3 &= \exists x \forall y (P(x) \to P(y))
\end{aligned}
$$

Aufgabe 2. Die Formel Φ_3 hat nach Skolemisierung und Negation die folgende Form:

$$\forall x (P(x) \wedge \neg P(f_y(x))$$

Zeige an dieser Formel wie der Gilmore-Algorithmus, der den Herbrandgedanken nutzt. Füge schrittweise erst eine Konstante c, dann $f_y(c)$ usw. ein und prüfe die so entstehenden Instanzen.

Aufgabe 3. Unifiziere folgende Mengen mit Hilfe des Algorithmus oder entscheide, dass sie nicht unifizierbar sind!

$$
\begin{aligned}
&\{(f(x,y,z), f(g(y,y), g(z,z), a)\} \\
&\{(f(G(a,x), y, h(a)), f(y,z,x))\} \\
&\{(F(h(x), h(y), x, y), F(z, g(z), a, z))\} \\
&\{(f(g(x), z, z), f(y,y,x)\} \\
&\{(f(x, h(y), y), f(g(z), x, z))\}
\end{aligned}
$$

Aufgabe 4. Definiere in OCaml:

```
#let unify_and_apply eqs =
let i = fullunify eqs in
let apply (t1,t2) = tsubst i t1,tsubst i t2 in
map apply eqs;;
```

Diese Funktion berechnet einen Unifikator und wendet ihn anschließend auf die ursprüngliche Liste an. Teste die Funktion an verschiedenen Instanzen. Verwende unter anderem

```
[<<|x_0|>>,<<|f(x_1,x_1)|>>;<<|x_1|>>,<<|f(x_2,x_2)|>>;
<<|x_2|>>,<<|f(x_3,x_3)|>>]
```

Interpretiere dieses Ergebnis hinsichtlich Laufzeit und Größe des Unifikators!