Unifikation

Unifikationsalgorithmus: $\texttt{unify}(C) =$

$$
\begin{array}{l}
\texttt{if}\ C\ \texttt{==}\ \emptyset\ \texttt{then}\ []\\
\texttt{else let}\ \{\tau_1 = \tau_2\} \cup C' = C\ \texttt{in}\\
\quad \texttt{if}\ \tau_1\ \texttt{==}\ \tau_2\ \texttt{then unify}(C')\\
\quad \texttt{else if}\ \tau_1\ \texttt{==}\ \alpha\ \texttt{and}\ \alpha \notin FV(\tau_2)\ \texttt{then unify}([\alpha \Rightarrow \tau_2]\, C') \circ [\alpha \Rightarrow \tau_2]\\
\quad \texttt{else if}\ \tau_2\ \texttt{==}\ \alpha\ \texttt{and}\ \alpha \notin FV(\tau_1)\ \texttt{then unify}([\alpha \Rightarrow \tau_1]\, C') \circ [\alpha \Rightarrow \tau_1]\\
\quad \texttt{else if}\ \tau_1\ \texttt{==}\ (\tau_1' \to \tau_1'')\ \texttt{and}\ \tau_2\ \texttt{==}\ (\tau_2' \to \tau_2'')\\
\qquad\quad \texttt{then unify}(C' \cup \{\tau_1' = \tau_2', \tau_1'' = \tau_2''\})\\
\quad \texttt{else}\ \textbf{fail}
\end{array}
$$

$\alpha \in FV(\tau)$ occur check, verhindert zyklische Substitutionen

Korrektheitstheorem

$\texttt{unify}(C)$ terminiert und gibt *mgu* für C zurück, falls C unifizierbar, ansonsten **`fail`**.

Beweis: Siehe Literatur