Dependency Pair: $\langle F(t^*), G(u^*)\rangle \in DP(P)$ gdw. es ex. $f(t^*) \equiv r \in E_P$ und $g(u^*)$ ist ein Teilterm von r mit $f \sim_P g$

Bsp: $\langle \mathsf{GE}(\mathsf{succ}(x), \mathsf{succ}(y)),\ \mathsf{GE}(x, y)\rangle$

$\langle \mathsf{SUBTRACT}(\mathsf{succ}(x), \mathsf{succ}(y)),\ \mathsf{SUBTRACT}(x, y)\rangle$

$\langle \mathsf{QUOT}(\mathsf{succ}(x), \mathsf{succ}(y)),\ \mathsf{QUOT}(\mathsf{subtract}(x, y), \mathsf{succ}(y))\rangle$

Kette: $\langle t_1, u_1\rangle, \langle t_2, u_2\rangle, \ldots$ ist *Kette* gdw.
es ex. $\sigma_1, \sigma_2, \ldots$ mit $\sigma_i(u_i) \Rightarrow_P^* \sigma_{i+1}(t_{i+1})$ für alle i.
Hierbei instantiiert σ_i t_i's Variablen mit Konstruktorgrundtermen.

$$
\begin{array}{ll}
\langle \mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)),\ \mathsf{Q}(\mathsf{sub}(x, y), \mathsf{s}(y))\rangle & \sigma_1 = \{x/\mathsf{s}^8(\mathcal{O}), y/\mathsf{s}^2(\mathcal{O})\} \\
\langle \mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)),\ \mathsf{Q}(\mathsf{sub}(x, y), \mathsf{s}(y))\rangle & \sigma_2 = \{x/\mathsf{s}^5(\mathcal{O}), y/\mathsf{s}^2(\mathcal{O})\} \\
\langle \mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)),\ \mathsf{Q}(\mathsf{sub}(x, y), \mathsf{s}(y))\rangle & \sigma_3 = \{x/\mathsf{s}^2(\mathcal{O}), y/\mathsf{s}^2(\mathcal{O})\}
\end{array}
$$

$$
\begin{array}{lcl}
\sigma_1(\mathsf{Q}(\mathsf{sub}(x, y), \mathsf{s}(y))) & \Rightarrow_P^* & \sigma_2(\mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y))) \\
\sigma_2(\mathsf{Q}(\mathsf{sub}(x, y), \mathsf{s}(y))) & \Rightarrow_P^* & \sigma_3(\mathsf{Q}(\mathsf{s}(x), \mathsf{s}(y)))
\end{array}
$$

Constraints für ge, if, subtract, quot

$$
\begin{array}{rcl}
\mathsf{GE}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{GE}(x, y) \\
\mathsf{SUBTRACT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{SUBTRACT}(x, y) \\
\mathsf{QUOT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{QUOT}(\mathsf{subtract}(x, y), \mathsf{succ}(y)) \\
\mathsf{subtract}(x, \mathcal{O}) & \succsim & x \\
\mathsf{subtract}(\mathcal{O}, \mathsf{succ}(y)) & \succsim & \mathcal{O} \\
\mathsf{subtract}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succsim & \mathsf{subtract}(x, y)
\end{array}
$$

Argument-Filterung mit $\mathcal{AF} = \{\mathsf{subtract}(x_1, x_2) \equiv \mathsf{subtract}'(x_1)\}$:

$$
\begin{array}{rcl}
\mathsf{GE}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{GE}(x, y) \\
\mathsf{SUBTRACT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{SUBTRACT}(x, y) \\
\mathsf{QUOT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{QUOT}(\mathsf{subtract}'(x), \mathsf{succ}(y)) \\
\mathsf{subtract}'(x) & \succsim & x \\
\mathsf{subtract}'(\mathcal{O}) & \succsim & \mathcal{O} \\
\mathsf{subtract}'(\mathsf{succ}(x)) & \succsim & \mathsf{subtract}'(x)
\end{array}
$$

Constraints für ge, if, subtract, quot

$$
\begin{array}{rcl}
\mathsf{GE}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{GE}(x, y) \\
\mathsf{SUBTRACT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{SUBTRACT}(x, y) \\
\mathsf{QUOT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{QUOT}(\mathsf{subtract}(x, y), \mathsf{succ}(y)) \\
\mathsf{subtract}(x, \mathcal{O}) & \succsim & x \\
\mathsf{subtract}(\mathcal{O}, \mathsf{succ}(y)) & \succsim & \mathcal{O} \\
\mathsf{subtract}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succsim & \mathsf{subtract}(x, y)
\end{array}
$$

Argument-Filterung mit $\mathcal{AF} = \{\mathsf{subtract}(x_1, x_2) \equiv x_1\}$:

$$
\begin{array}{rcl}
\mathsf{GE}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{GE}(x, y) \\
\mathsf{SUBTRACT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{SUBTRACT}(x, y) \\
\mathsf{QUOT}(\mathsf{succ}(x), \mathsf{succ}(y)) & \succ & \mathsf{QUOT}(x, \mathsf{succ}(y)) \\
x & \succsim & x \\
\mathcal{O} & \succsim & \mathcal{O} \\
\mathsf{succ}(x) & \succsim & x
\end{array}
$$

$$\mathcal{U}(E_P, \mathsf{QUOT}(\mathsf{subtract}(x,y), \mathsf{succ}(y)))$$

$$\begin{aligned}
&= \mathcal{U}(E_P, \mathsf{subtract}(x,y)) \cup \mathcal{U}(E_P, \mathsf{succ}(y)) \\
&= \mathcal{U}(E_P, \mathsf{subtract}(x,y)) \\
&= (E_P)_{\mathsf{subtract}} \cup \bigcup_{l \equiv r \in (E_P)_{\mathsf{subtract}}} \mathcal{U}(E_P \setminus (E_P)_{\mathsf{subtract}}, r) \cup \mathcal{U}(..., x) \cup \mathcal{U}(..., y) \\
&= (E_P)_{\mathsf{subtract}} \cup \bigcup_{l \equiv r \in (E_P)_{\mathsf{subtract}}} \mathcal{U}(E_P \setminus (E_P)_{\mathsf{subtract}}, r) \\
&= (E_P)_{\mathsf{subtract}} \cup \mathcal{U}(\ldots, x) \cup \mathcal{U}(\ldots, \mathcal{O}) \cup \mathcal{U}(E_P \setminus (E_P)_{\mathsf{subtract}}, \mathsf{subtract}(x,y)) \\
&= (E_P)_{\mathsf{subtract}}
\end{aligned}$$

$$\begin{aligned}
= \;&\{\mathsf{subtract}(x, \mathcal{O}) \equiv x, \\
&\;\mathsf{subtract}(\mathcal{O}, \mathsf{succ}(y)) \equiv \mathcal{O}, \\
&\;\mathsf{subtract}(\mathsf{succ}(x), \mathsf{succ}(y)) \equiv \mathsf{subtract}(x,y)\}.
\end{aligned}$$

Constraints für even und odd

$$\begin{array}{rcl} \mathsf{EVEN}(\mathsf{succ}(x)) & \succ & \mathsf{ODD}(x) \\ \mathsf{ODD}(\mathsf{succ}(x)) & \succ & \mathsf{EVEN}(x) \end{array}$$

$$\begin{array}{rcl} \mathsf{even}(\mathcal{O}) & \succsim & \mathsf{true} \\ \mathsf{even}(\mathsf{succ}(x)) & \succsim & \mathsf{odd}(x) \\ \mathsf{odd}(\mathcal{O}) & \succsim & \mathsf{false} \\ \mathsf{odd}(\mathsf{succ}(x)) & \succsim & \mathsf{even}(x) \end{array}$$

Argument-Filterung

$\mathcal{AF} = \{\mathsf{EVEN}(x_1) \equiv x_1, \mathsf{ODD}(x_1) \equiv x_1, \mathsf{even}(x_1) \equiv x_1, \mathsf{odd}(x_1) \equiv x_1\}$

$$\begin{array}{rcl} \mathsf{succ}(x) & \succ & x \\ \mathsf{succ}(x) & \succ & x \end{array}$$

$$\begin{array}{rcl} \mathcal{O} & \succsim & \mathsf{true} \\ \mathsf{succ}(x) & \succsim & x \\ \mathcal{O} & \succsim & \mathsf{false} \\ \mathsf{succ}(x) & \succsim & x \end{array}$$

Algorithmus TERM(P)

Eingabe: Ein Programm P

Ausgabe: "Terminiert" oder "Keine Aussage".
Falls "Terminiert" ausgegeben wird, dann terminiert P.

1. Für alle Argument-Filterungssysteme $\mathcal{AF}$ für die Signatur, die aus der zu P gehörenden Signatur durch die Erweiterung um die entsprechenden Tupelsymbole entsteht:

   2. Für alle lexikographischen Pfadordnungen $\succ_{lpo}$
   (oder alle rekursiven Pfadordnungen mit Status $\succ_{rpos}$):

      3. Falls $t{\downarrow_{\mathcal{AF}}} \succ_{lpo} u{\downarrow_{\mathcal{AF}}}$ für alle $\langle t, u \rangle$ aus $DP(P)$ und
      $l{\downarrow_{\mathcal{AF}}} \succeq_{lpo} r{\downarrow_{\mathcal{AF}}}$ für alle $l \equiv r$ aus $\bigcup_{\langle t,u \rangle \in DP(P)} \mathcal{U}(E_P, u)$,
      dann gib aus "Terminiert" und breche den Algorithmus ab.

4. Gib aus "Keine Aussage".

Scheitern von Dependency Pairs mit Simplifikationsordnungen

$$
\begin{array}{lcl}
\multicolumn{3}{l}{\textbf{function}\ \ \mathsf{fail} : \mathsf{bool} \times \mathsf{bool} \times \mathsf{bool} \to \mathsf{bool}} \\
\quad \mathsf{fail}(\mathsf{false}, b, c) & \equiv & \mathsf{false} \\
\quad \mathsf{fail}(\mathsf{true}, \mathsf{true}, c) & \equiv & \mathsf{true} \\
\quad \mathsf{fail}(\mathsf{true}, \mathsf{false}, c) & \equiv & \mathsf{fail}(c, c, c)
\end{array}
$$

Es existiert kein $\mathcal{AF}$ und keine Simplifikationsordnung $\succ$ mit

$$\mathsf{FAIL}(\mathsf{true}, \mathsf{false}, c) \ \succ \ \mathsf{FAIL}(c, c, c).$$