

Zusammenfassung Term Rewriting aAT

© Tim Baumann, <http://timbaumann.info/uni-spicker>

Dies ist eine übersetzte Zusammenfassung des Buches Term Rewriting and All That von Franz Baader und Tobias Nipkow.

Abstrakte Reduktionssysteme

Def. Ein **abstraktes Reduktionssystem** ist ein Tupel (A, \rightarrow) , wobei $\rightarrow \in A \times A$ eine Relation auf A ist.

Def.

$\xrightarrow{0} := \{(a, a) \mid a \in A\}$	Identität
$\xrightarrow{i+1} := \xrightarrow{i} \circ \rightarrow$	$(i+1)$ -fache Komposition, $i \geq 0$
$\leftarrow := \{(t, s) \mid (s, t) \in \rightarrow\}$	Inverse Relation
$\xrightarrow{*} := (\rightarrow) \cup (\xrightarrow{0})$	refl. Hülle
$\xrightarrow{+} := \bigcup_{i \geq 0} (\xrightarrow{i})$	refl. trans. Hülle
$\xrightarrow{+} := \bigcup_{i \geq 1} (\xrightarrow{i})$	refl. trans. Hülle
$\leftrightarrow := \rightarrow \cup \leftarrow$	symm. Hülle
$\leftrightarrow^* := (\leftrightarrow)^*$	refl. trans. symm. Hülle

Def. Sei $x \in A$ ein Term.

- Der Term x heißt **reduzibel**, falls ein $y \in A$ mit $x \rightarrow y$ existiert,
- **irreduzibel** (oder in **Normalform**) falls x nicht reduzibel ist.
- Ein Term $y \in A$ heißt **Normalform** von x , falls $x \xrightarrow{*} y$ und y irreduzibel ist.
- Eine Term y heißt **direkter Nachfolger** von x , falls $x \rightarrow y$.
- Eine Term y heißt **Nachfolger** von x , falls $x \xrightarrow{+} y$.
- x und y heißen *joinable*, notiert $x \downarrow y$, falls $\exists z : x \xrightarrow{*} z \leftarrow^* y$.

Def. Eine Reduktion \rightarrow heißt

Church-Rosser : $\iff x \leftrightarrow^* y \implies x \downarrow y$
konfluent : $\iff y_1 \leftarrow^* y \xrightarrow{*} y_2 \implies y_1 \downarrow y_2$
semi-konfluent : $\iff y_1 \leftarrow y \xrightarrow{*} y_2 \implies y_1 \downarrow y_2$
terminierend : \iff es gibt keine unendlich absteigende Kette $x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \dots$ (auch: *noethersch*)
normalisierend : \iff jeder Term besitzt eine Normalform
konvergent : \iff konfluent \wedge normalisierend

Lem. Für eine Reduktion \rightarrow sind äquivalent:

- \rightarrow ist Church-Rosser
- \rightarrow ist konfluent
- \rightarrow ist semi-konfluent

Lem. Ist die Reduktion \rightarrow konfluent/terminierend/konvergent, so besitzt jeder Term höchstens/mindestens/genau eine Normalform.

Notation. Falls x eine NF y besitzt, so schreibe $x := \downarrow y$.

Thm. Ist \rightarrow konvergent, so gilt $x \leftrightarrow^* y \iff x \downarrow = y \downarrow$.

Bem. Dies liefert einen einfachen Algorithmus, um $x \leftrightarrow^* y$ zu entscheiden: Reduziere die Terme x und y zu Normalformen $x \downarrow$ bzw. $y \downarrow$ und vergleiche diese.

Terminierungsbeweise

Lem. \rightarrow ist terminierend $\iff \rightarrow$ ist eine Wohlordnung

Def. Eine Relation \rightarrow heißt

- **endlich verzweigend**, falls jeder Term nur endlich viele direkte Nachfolger besitzt,
- **global endlich**, falls jeder Term nur endl. viele Nachfolger hat,
- **azyklisch**, falls kein Term a mit $a \xrightarrow{+} a$ existiert.

Lem.

- Eine endlich verzweigende Relation ist global endlich, falls sie terminierend ist.
- Eine azykl. Relation ist terminierend, falls sie global endlich ist.

Lem. Sei (A, \rightarrow) ein Reduktionssystem und $(B, >)$ eine wohlgeordnete Menge. Gibt es eine streng monotone Abbildung $\varphi : A \rightarrow B$, so ist A terminierend.

Lem. Ein endlich verzweigendes Reduktionssystem (A, \rightarrow) ist genau dann terminierend, falls es eine streng monotone Abbildung $\varphi : (A, \rightarrow) \rightarrow (\mathbb{N}, >)$ gibt.

Def. Seien $(A_i, >_i)_{i=1, \dots, n}$ geordnete Mengen. Die **lexikalische Ordnung** $>_{\text{lex}}$ auf $A_1 \times \dots \times A_n$ ist definiert durch

$(x_1, \dots, x_n) >_{\text{lex}} (y_1, \dots, y_n) : \iff \exists k \leq n : (\forall i < k : x_i = y_i) \wedge x_k <_k y_k$.

Lem. Ist $>$ eine strikte (Wohl-) Ordnung, so auch $>_{\text{lex}}$.

Def. Eine *Multimenge* M über einer Menge A ist eine Abbildung $M : A \rightarrow \mathbb{N}$. Sie ist endlich, falls $\sum_{a \in A} M(a) < \infty$.

Notation. $\mathcal{M}(A) := \{ \text{Multimengen über } A \}$
 $a \in M : \iff M(a) \geq 1$

Def. Die *Differenz* von Multimengen $M, N \in \mathcal{M}(A)$ ist $M - N \in \mathcal{M}(A)$ mit $(M - N)(a) := \max\{0, M(a) - N(a)\}$.

Def. Sei $>$ eine strikte Ordnung auf A . Die **Multimengenordnung** $>_{\text{mul}}$ auf $\mathcal{M}(A)$ ist dann definiert durch

$M >_{\text{mul}} N : \iff M \neq N \wedge \forall n \in N - M : \exists m \in M - N : m > n$.

Lem. Ist $>$ eine strikte (Wohl-) Ordnung, so auch $>_{\text{mul}}$.

Konfluenzbeweise

Def. Eine Relation \rightarrow

- heißt **lokal konfluent**, falls $y_1 \leftarrow y \rightarrow y_2 \implies y_1 \downarrow y_2$.
- heißt **stark konfluent**, falls $y_1 \leftarrow y \rightarrow y_2 \implies \exists z : y_1 \xrightarrow{*} z \xleftarrow{*} y_2$.
- besitzt die **Diamant-Eigenschaft**, falls

$$y_1 \leftarrow y \rightarrow y_2 \implies \exists z : y_1 \rightarrow z \leftarrow y_2.$$

Lem. Falls $\rightarrow_1 \leq \rightarrow_2 \leq \xrightarrow{*}_1$, so gilt $\xrightarrow{*}_1 = \xrightarrow{*}_2$. Ist zusätzlich \rightarrow_2 (stark) konfluent, so auch \rightarrow_1 .

Lem.

- Stark konfluente Relationen sind konfluent.
- Eine terminierende Rel. ist konfluent, falls sie lokal konfluent ist.

Def. Zwei Relationen \rightarrow_1 und \rightarrow_2 auf A

- **kommutieren**, falls $y_1 \leftarrow^*_1 x \xrightarrow{*}_2 y_2 \implies \exists z : y_1 \xrightarrow{*}_2 z \xleftarrow{*}_1 y_2$.
- **kommutieren stark**, falls

$$y_1 \leftarrow_1 x \rightarrow_2 y_2 \implies \exists z : y_1 \xrightarrow{=}_2 z \xleftarrow{*}_1 y_2.$$

- besitzen die **Kommutierender-Diamant-Eigenschaft**, falls

$$y_1 \leftarrow_1 x \rightarrow_2 y_2 \implies \exists z : y_1 \rightarrow_2 z \leftarrow_1 y_2.$$

Lem. Angenommen, \rightarrow_1 und \rightarrow_2 sind konfluent und kommutieren. Dann ist auch $\rightarrow_1 \cup \rightarrow_2$ konfluent.

Universelle Algebra

Def. Eine **Signatur** Σ ist eine Menge von *Funktionssymbolen* zusammen mit einer Aritätsabbildung $\text{arity} : \Sigma \rightarrow \mathbb{N}$.

Notation. $\Sigma^{(n)} := \text{arity}^{-1}(n)$

Def. Sei Σ eine Signatur und X eine Menge von Variablen (d. h. es gilt $X \cap \Sigma = \emptyset$). Die Menge $T(\Sigma, X)$ der **Σ -Terme über X** ist induktiv definiert durch

- $X \subseteq T(\Sigma, X)$
- $\forall f \in \Sigma^{(n)}, t_1 \in T(\Sigma, X), \dots, t_n \in T(\Sigma, X) : f(t_1, \dots, t_n) \in T(\Sigma, X)$

Bem. Falls $X \subseteq Y$, $Y \cap \Sigma = \emptyset$, so gilt $T(\Sigma, X) \subseteq T(\Sigma, Y)$.

Def. Terme t ohne freie Variablen (d. h. $t \in T(\Sigma, \emptyset)$) heißen **Grundterme** oder **geschlossene Terme**.

Def. Die Menge der **Positionen** $\text{Pos}(s)$ eines Terms $s \in T(\Sigma, X)$ ist folgende Menge von Listen von natürlichen Zahlen

- Falls $s = x \in X$: $\text{Pos}(s) := \{\epsilon\}$
- Falls $s = f(s_1, \dots, s_n)$: $\text{Pos}(s) := \{\epsilon\} \cup \bigcup_{i=1}^n \{ip \mid p \in \text{Pos}(s_i)\}$

Def. Die **Größe** eines Terms $s \in T(\Sigma, X)$ ist $|s| := |\text{Pos}(s)|$.

Def. Der **Subterm** $s|_p$ an der Position $p \in \text{Pos}(s)$ eines Terms s ist

$$s|_\epsilon := s, \quad f(s_1, \dots, s_n)|_{iq} := s_i|_q.$$

Die **Ersetzung** $s[t]_p$ von $s|_p$ durch einen Term $t \in T(\Sigma, X)$ ist

$$s[t]_\epsilon := t, \quad f(s_1, \dots, s_n)[t]_{iq} := s_i[t]_q.$$

Def. Die **Menge der Variablen** in $s \in T(\Sigma, X)$ ist

$$\text{Var}(s) := \{x \in X \mid \exists p \in \text{Pos}(s) : s|_p = x\}.$$

Bem. Für jeden Term $t \in T(\Sigma, X)$ gilt $t \in T(\Sigma, \text{Var}(t))$.

Def. Sei Σ eine Signatur und V eine abzählbar unendliche Menge von Variablen. Eine $T(\Sigma, V)$ -**Ersetzung** ist eine Abbildung $\sigma : V \rightarrow T(\Sigma, V)$, für die gilt:

$$\text{Dom}(\sigma) := \{v \in V \mid \sigma(v) \neq v\}$$

ist endlich. Die Menge der $T(\Sigma, V)$ -Ersetzungen ist $\text{Sub}(T(\Sigma, V))$. Wir können σ ausdehnen zu einer Abb. $\hat{\sigma} : T(\Sigma, V) \rightarrow T(\Sigma, V)$ durch

$$\hat{\sigma}(v) := \sigma(v), \quad \hat{\sigma}(f(s_1, \dots, s_n)) := f(\hat{\sigma}(s_1), \dots, \hat{\sigma}(s_n)).$$

Die **Komposition** zweier Ersetzungen σ und τ ist $\sigma \circ \tau := \hat{\sigma} \circ \tau$.

Def. Eine **Σ -Identität** ist ein Paar $(s, t) \in T(\Sigma, V) \times T(\Sigma, V)$, auch geschrieben $s \approx t$.

Def. Die **Reduktionsrelation** \rightarrow_E zu einer Menge E von Σ -Identitäten ist

$$s \rightarrow_E t : \iff \exists (l \approx r) \in E, p \in \text{Pos}(s), \sigma \in \text{Sub}(T(\Sigma, V)) : \\ s|_p = \sigma(l) \wedge t = s[\sigma(r)]_p.$$

Def. Eine Relation \equiv auf $T(\Sigma, V)$ heißt

- **abgeschlossen unter Ersetzungen**, falls $s = t \implies \sigma(s) = \sigma(t)$
- **abgeschlossen unter Σ -Operationen**, falls

$$s_1 \equiv t_1, \dots, s_n \equiv t_n \implies f(s_1, \dots, s_n) \equiv f(t_1, \dots, t_n)$$

- **kompatibel mit Σ -Operationen**, falls

$$s \equiv t \implies f(s_1, \dots, s_{i-1}, s, s_{i+1}, s_n) \equiv f(s_1, \dots, s_{i-1}, t, s_{i+1}, s_n)$$

- **kompatibel mit Σ -Kontexten**, falls

$$s \equiv s' \implies t[s]_p \equiv t[s']_p$$

- **Umschreibungsrelation**, falls sie kompatibel mit Σ -Operationen und abgeschlossen unter Ersetzungen ist.

Lem. Es sind äquivalent:

- \equiv ist kompatibel mit Σ -Operationen
- \equiv ist kompatibel mit Σ -Kontexten

Ist \equiv reflexiv und transitiv, so ist außerdem äquivalent:

- \equiv ist abgeschlossen unter Σ -Operationen

Thm. Sei E eine Menge von Σ -Identitäten.

- \rightarrow_E , $\overset{+}{\rightarrow}_E$ und $\overset{*}{\rightarrow}_E$ sind Umschreibungsrelationen.
- Die Relation $\overset{*}{\leftrightarrow}_E$ ist die kleinste Äquivalenzrelation, die E enthält und abg. ist unter Ersetzungen und Σ -Operationen.

Def. Eine **Σ -Algebra** \mathcal{A} besteht aus

- einer *Trägermenge* A und
- einer Abbildung $f^{\mathcal{A}} : A^n \rightarrow A$ für alle $f \in \Sigma^{(n)}$.

Bsp. $T(\Sigma, V)$ ist eine Σ -Algebra mit

$$f^{T(\Sigma, V)} : T(\Sigma, V)^n \rightarrow T(\Sigma, V), \quad (t_1, \dots, t_n) \mapsto f(t_1, \dots, t_n).$$

- Eine **Σ -Subalgebra** von A ist eine Teilmenge $B \subset A$, sodass $f^{\mathcal{A}}(b_1, \dots, b_n) \in B$ für alle $f \in \Sigma^{(n)}$ und $b_1, \dots, b_n \in B$.
- Die von $X \subseteq A$ **erzeugte Σ -Subalgebra** ist die kleinste Σ -Subalgebra, die X enthält.

Def. Ein *Homomorphismus* ϕ zwischen Σ -Algebren \mathcal{A} und \mathcal{B} (mit Trägermengen A bzw. B) ist eine Abbildung $\phi : A \rightarrow B$, sodass

$$\phi(f^{\mathcal{A}}(a_1, \dots, a_n)) = f^{\mathcal{B}}(\phi(a_1), \dots, \phi(a_n)).$$

Bem. Damit bilden Σ -Algebren eine Kategorie.

Def. Eine Äquivalenzrelation \equiv auf A heißt **Kongruenz** auf \mathcal{A} , falls

$$a_1 \equiv b_1, \dots, a_n \equiv b_n \implies f^{\mathcal{A}}(a_1, \dots, a_n) \equiv f^{\mathcal{A}}(b_1, \dots, b_n).$$

Lem/Def. Ist \equiv eine Äquivalenz, so wird A/\equiv mit

$$f^{\mathcal{A}/\equiv}([a_1], \dots, [a_n]) := [f^{\mathcal{A}}(a_1, \dots, a_n)]$$

eine Σ -Algebra, die **Quotientenalgebra** \mathcal{A}/\equiv .

Lem. Die Kategorie der Σ -Algebren enthält kleine Limiten.

Def. Eine Σ -Algebra heißt **frei**, falls sie isomorph ist zu $F(X) := T(\Sigma, X)$ für eine Menge X von Variablen.

Bem. Diese Setzung definiert einen Funktor $F : \mathbf{Set} \rightarrow \Sigma\text{-}\mathbf{Alg}$.

Lem. $F \dashv U$, wobei $U : \mathbf{Set} \rightarrow \Sigma\text{-}\mathbf{Alg}$ der Vergissfunctor ist.

Kor. $F(\emptyset) = T(\Sigma, \emptyset)$ ist das initiale Objekt in $\Sigma\text{-}\mathbf{Alg}$.

- Eine Σ -Identität $s \approx t$ **gilt in einer Σ -Algebra** \mathcal{A} , falls für alle Homomorphismen $\phi : T(\Sigma, V) \rightarrow \mathcal{A}$ gilt: $\phi(s) = \phi(t)$.
- \mathcal{A} ist ein **Modell** einer Menge E von Σ -Algebren (notiert $\mathcal{A} \models E$), falls jede Identität aus E in \mathcal{A} gilt.
- Die Subkategorie von $\Sigma\text{-}\mathbf{Alg}$ der Modelle von E heißt *durch E definierte Σ -Varietät* $\mathcal{V}(E)$.

- Die Identität $s \approx t$ ist eine **semantische Konsequenz** von E (notiert $E \models s \approx t$), falls $s \approx t$ in allen $\mathcal{A} \in \mathcal{V}(E)$ gilt.
- $\approx_E := \{(s, t) \mid E \models s \approx t\}$ heißt von E **induzierte Theorie**.

Def. Eine Relation \equiv auf $T(\Sigma, V)$ heißt **voll invariant**, falls $s \equiv t \implies \phi(s) \equiv \phi(t)$ für alle Mor. $\phi : T(\Sigma, V) \rightarrow T(\Sigma, V)$.

Lem. \approx_E ist eine voll invariante Kongruenz.

Lem/Def. Es sind äquivalent:

- E heißt **trivial**
- $\approx_E = T(\Sigma, V) \times T(\Sigma, V)$
- $x \approx_E y$ gilt für Variablen $x, y \in V$, $x \neq y$
- $\mathcal{V}(E)$ besteht aus Algebren der Kardinalität ≤ 1 .

Thm. Sei V eine abzählbar unendliche Menge von Variablen.

- $T(\Sigma, V)/\approx_E$ ist eine freie Algebra in $\mathcal{V}(E)$ mit erz. Menge V/\approx_E . Falls E nicht trivial ist, so ist V/\approx_E abzählbar unendlich.
- $T(\Sigma, V)/\approx_E \models s \approx t \iff s \approx_E t$

Def. Die durch E **induzierte induktive Theorie** ist

$$\approx_E^I := \{(s, t) \mid T(\Sigma, \emptyset) \models s \approx t\} \subseteq T(\Sigma, V) \times T(\Sigma, V).$$

Bem. $\approx_E \subseteq \approx_E^I$

Umformulierung. Die Relation $\overset{*}{\leftrightarrow}_E$ ist die kleinste voll invariante Kongruenz auf $T(\Sigma, V)$, die E enthält.

Lem. Für eine voll invariante Kongruenz \equiv auf $T(\Sigma, V)$ gilt:

$$E \subseteq \equiv \implies \approx_E \subseteq \equiv.$$

Kor (Birkhoffs Lemma). $\overset{*}{\leftrightarrow}_E = \approx_E$

Thm. Für eine Klasse \mathcal{K} von Σ -Algebren sind äquivalent:

- \mathcal{K} ist eine Varietät, d. h. $\mathcal{K} = \mathcal{V}(E)$ für eine Menge E von Identitäten.
- \mathcal{K} ist abgeschlossen unter dem Bilden von Unteralgebren, Bildalgebren und direkten Produkten.

Gleichheitsprobleme

Def. Sei E eine Menge von Identitäten. Eine Gleichheit $s \approx t$ heißt

- **gültig** in E , falls $s \approx_E t$,
- **erfüllbar** in E , falls es eine Ersetzung σ mit $\sigma(s) \approx_E \sigma(t)$ gibt.

Problem (matching problem). Gegeben Terme s und l , gibt es eine Ersetzung σ , sodass $\sigma(s) = l$?

Thm. Ist E endlich und \rightarrow_E konvergent, so ist \approx_E entscheidbar.

Algorithmus. Seien x und y gegeben. Wegen der Endlichkeit von E sind $x \downarrow$ und $y \downarrow$ berechenbar. Es gilt $x \approx_E y \iff x \downarrow = y \downarrow$.

- Def.** • **Wortproblem:** Gegeben $x, y \in T(\Sigma, V)$, gilt $x \approx_E y$?
- **Grundwortproblem:** Gegeben $x, y \in T(\Sigma, \emptyset)$, gilt $x \approx_E y$?

Bem. Das Wortproblem ist im Allgemeinen unentscheidbar, denn:

- Man kann den turingvollständigen SKI-Kalkül als Reduktionssystem durch Angabe einer Menge von Gleichheiten spezifizieren.
- Gleichheit von Programmen ist unentscheidbar.

Def. • Eine **Umschreibungsregel** ist eine Identität $l \approx r$ bei der s keine Variable ist und $\text{Var}(l) \supseteq \text{Var}(r)$.

- Ein **Termumschreibungssystem** ist eine Menge von Umschreibungsregeln.

Bem. Die zwei Bedingungen für Umschreibungsregeln sind notwendig (aber nicht hinreichend) dafür, dass Termumschreibungssysteme terminierend sind.

Die kongruente Hülle

Def. Die **kongruente Hülle** $\text{CC}(E)$ von $E \subseteq T(\Sigma, V) \times T(\Sigma, V)$ ist die kleinste Kongruenzrelation, die \equiv enthält.

Bem. $(s, t) \in \text{CC}(E)$ gilt genau dann, wenn die Aussage aus folgenden Inferenzregeln herleitbar ist:

$$\frac{}{(t, t) \in \text{CC}(E)} \quad \frac{(t, s) \in \text{CC}(E)}{(s, t) \in \text{CC}(E)} \quad \frac{(r, s) \in \text{CC}(E) \quad (s, t) \in \text{CC}(E)}{(r, t) \in \text{CC}(E)} \\ \frac{(s, t) \in E}{(s, t) \in \text{CC}(E)} \quad \frac{f \in \Sigma^{(n)} \quad (s_1, t_1) \in \text{CC}(E), \dots, (s_n, t_n) \in \text{CC}(E)}{(f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \in \text{CC}(E)}$$

Def. Eine Id. $l \approx r$ heißt **Grundidentität**, falls $\text{Var}(l) = \text{Var}(r) = \emptyset$.

Notation. Sei G im Folgenden eine Menge von Grundidentitäten.

Lem. $\text{CC}(G) = \approx_G$

Def. Die Menge der **Unterterme** ist

$$\text{Subterms}(t) := \{t|_p \mid p \in \text{Pos}(t)\} \quad \text{für } t \in T(\Sigma, V) \text{ bzw.} \\ \text{Subterms}(G) := \bigcup_{l \approx r} \text{Subterms}(l) \cup \text{Subterms}(r).$$

Thm. Fixiere zwei Terme $s, t \in T(\Sigma, V)$. Setze

$$S := \text{Subterms}(s) \cup \text{Subterms}(t) \cup \text{Subterms}(G).$$

Es gilt $G \subseteq S \times S$. Es sei $\text{CC}_S(G)$ die kongruente Hülle von G innerhalb von $S \times S$. Dann gilt:

$$\text{CC}_S(G) = \approx_G \cap (S \times S).$$

Kor. Das Wortproblem ist für endliche Mengen G von Grundidentitäten entscheidbar.

Beweisidee. Seien s und t gegeben. Berechne die endliche Menge $\text{CC}_S(G)$. Es gilt dann: $s \approx_G t \iff (s, t) \in \text{CC}_S(G)$.

Bem. Dies liefert einen Entscheidungsalgorithmus mit polynomieller Laufzeit in G , s und t .

Algorithmus. Effiziente Realisierung:

- Repräsentiere die Termmenge S als gerichteter Graph, wobei jeder Knoten v mit einem Symbol $f \in \Sigma$ beschriftet ist und dessen Auskanten mit $i = 1, \dots, \text{arity}(f)$ nummeriert sind.
- Wir repräsentieren Identifikationen von Knoten im Graph über Zeiger wie in der Union-Find-Datenstruktur. Wir definieren $u \sim v : \iff \text{FIND}(u) = \text{FIND}(v)$ für Knoten u und v .

```

1: function MERGE(u, v)
2:   if u ≠ v then
3:     P := PRED(u), Q := PRED(v)
4:     UNION(u, v)
5:     for (p, q) ∈ P × Q do
6:       if p ≠ q ∧ CONGRUENT(p, q) then
7:         MERGE(p, q)
8: function CONGRUENT(p = f(p1, ..., pn), q = g(q1, ..., qm))
9:   if f ≠ g ∈ Σ then return false
10:  for i = 1, ..., n do
11:    if pi ≠ qi then return false
    return true

```

- Rufe zu Beginn des Algorithmus $\text{MERGE}(l, r)$ für alle Grundidentitäten $(l \approx r) \in G$ auf.
- Das Ergebnis ist nun $s \sim t$.

Syntaktische Unifikation

Def. Eine Substitution σ heißt **allgemeiner** (notiert $\sigma \lesssim \sigma'$) als σ' , falls eine Substitution δ mit $\sigma' = \delta\sigma$ existiert.

Lem. \lesssim ist eine Quasiordnung

Def. Eine **Umbenennung** ist eine Ersetzung ρ mit $\text{im}(\rho) \subseteq V$ ($\implies \text{im}(\rho) = V$).

Lem. $\sigma \lesssim \sigma' \wedge \sigma' \lesssim \sigma \iff \exists \text{ Umbenennung } \rho : \sigma = \rho\sigma'$

Def. Ein **Unifikationsproblem** ist gegeben durch eine endliche Menge von Gleichungen

$$S = \{s_1 \stackrel{?}{=} t_1, \dots, s_n \stackrel{?}{=} t_n\}.$$

Eine *Lösung* von S ist eine Ersetzung σ mit $\hat{\sigma}(s_i) = \hat{\sigma}(t_i)$ für $i = 1, \dots, n$. Notation: $\mathcal{U}(S) := \{ \text{Lösungen von } S \}$

Gesucht. Eine **allgemeinste Lösung** von S , das ist ein bezüglich \lesssim kleinstes Element in $\mathcal{U}(S)$.

Thm. Hat ein Unifikationsproblem eine Lösung, so hat es auch eine idempotente, allgemeinste Lösung.

Def. Ein Unifikationsproblem $S = \{x_1 \stackrel{?}{=} t_1, \dots, x_n \stackrel{?}{=} t_n\}$ ist in **gelöster Form**, falls x_1, \dots, x_n paarweise verschieden Variablen sind, die nicht in den Termen t_1, \dots, t_n auftreten. In diesem Fall ist

$$\vec{S} := \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}.$$

Lem. Sei S in gelöster Form. Dann gilt:

- $\forall \sigma \in \mathcal{U}(S) : \sigma = \sigma\vec{S}$

- \vec{S} ist eine idempotente, allgemeinste Lösung von S .

Algorithmus ($\text{UNIFY}(S)$). Wende wiederholt folgende Transformationsregeln (in beliebiger Reihenfolge) auf S an:

Delete	$\{t \stackrel{?}{=} t\} \sqcup S$	\rightsquigarrow	S
Decompose	$\{f(\vec{s}) \stackrel{?}{=} f(\vec{t})\}$	\rightsquigarrow	$\{s_1 \stackrel{?}{=} t_1, \dots, s_n \stackrel{?}{=} t_n\} \cup S$
Orient	$\{t = x\} \sqcup S$	\rightsquigarrow	$\{x = t\} \cup S$ falls $t \notin V$
Eliminate	$\{x \stackrel{?}{=} t\} \sqcup S$	\rightsquigarrow	$\{x \stackrel{?}{=} t\} \cup [t/x]S$ falls $x \in \text{Var}(S) \setminus \text{Var}(t)$

Wenn keine Transformationsregel mehr angewandt werden kann, so

- gib \vec{T} zurück, falls die nach Anwendung aller Transformationen erhaltene Gleichungsmenge T in gelöster Form ist,
- ansonsten gib \perp zurück.

Lem. Falls $S \rightsquigarrow T$, so gilt $\mathcal{U}(S) = \mathcal{U}(T)$.

Thm. $\text{UNIFY}(S)$ ist korrekt:

- Der Algorithmus terminiert für alle Eingaben.
- Die Ausgabe ist eine idempotente, allgemeinste Lösung von S oder \perp , falls S keine Lösung besitzt.

Bem. Folgende Regeln bewirken einen frühen Abbruch:

Clash	$\{f(\vec{s}) \stackrel{?}{=} g(\vec{t})\} \sqcup S$	\rightsquigarrow	\perp	falls $f \neq g$
Occurs-Check	$\{x \stackrel{?}{=} t\} \sqcup S$	\rightsquigarrow	\perp	falls $x \in \text{Var}(t)$ und $x \neq t$

Bem. Naive Implementierungen von Unifikation benötigen exponentielle Zeit. Es gibt einen Algorithmus auf Termgraphen, der nur (fast) lineare Zeit benötigt.