Cheat Sheet Equational Logic (Spring 2013)

- The material given here summarizes those notions from the course's textbook [1] that occur frequently. The goal is to have them at hand, as a quick reminder at a single glance. References to [1] are given throughout.
- Contrary to textbook usage, we use gray text color to indicate side-remarks, i.e. additions that are formally necessary but clear in principle. We hope to keep your learning focussed by this device.
- Updates of the cheat sheet will be posted as we go along, so perhaps do not print it prematurely.

Enjoy rewriting terms!

Terms

Inductive Construction

Def. (3.1.1) *signature* Σ: set of function symbols, each with associated non-negative *arity* (number of arguments). The subset of symbols with arity n is denoted by $\Sigma^{(n)}$. *constants* are nullary functions $\Sigma^{(0)}$.

signature Σ
arity $\Sigma^{(n)}$
constants

Def. (3.1.2) For signature Σ and *variables* X (with $\Sigma \cap X = \emptyset$ and X countably infinite), define the Σ-*terms* over X (written $T(\Sigma, X)$) inductively

- Variables are terms: $X \subseteq T(\Sigma, X)$
- Function application yields new terms:
  for $t_1 \dots t_n \in T(\Sigma, X)$ also $f(t_1 \dots t_n) \in T(\Sigma, X)$

variables
terms $T(\Sigma, X)$

Positions: Denoting Subterms

Def. (3.1.3) The *positions* $\mathcal{P}os(s)$ of a term are sequences of integers, defined inductively along the term structure.

positions $\mathcal{P}os(s)$

- Variable terms $s \in X$ have a single position $\mathcal{P}os(s) := \{\varepsilon\}$
- Function application with arity n pre-prends child-positions with the child's index:

$$\mathcal{P}os(f(s_1 \dots s_n)) := \{\varepsilon\} \cup \bigcup_{i=1}^{n} \{ip | p \in \mathcal{P}os(s_i)\}$$

Def. (3.1.3) The *subterm* at position p in term s, written $s|_p$ is defined by following the position-indexes

subterm $s|_p$

- $s|_\varepsilon := s$
- $f(s_1 \dots s_n)|_{iq} := s_i|_q$

Special Positions

Def. (3.1.3) The position ε is called *root position*, the function or variable found there is the *root symbol*.

root position
root symbol

Def. (3.1.3) Term *variables occurring in* s are written $\mathcal{V}ar(s)$.
and defined by $\mathcal{V}ar(s) := \{x \in X | \exists p \in \mathcal{P}os(s).\ s|_p = x\}$

variables occurring in

Def. (3.1.3) A position with $p \in \mathcal{P}os(s)$ with $s|_p \in X$ is a *variable position*.

variable position

Manipulating Terms

Subterm Replacement

Def. (3.1.3) *Replacing* the subterm at p in s by t is written $s[t]_p$ and defined by following the position and then replacing the subterm found there by the given term. By induction on the position:

Replacing $s[t]_p$

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

Instantiation and Substitutions

Basics

Def. (3.1.6) A $T(\Sigma, V)$-*substitution* is a function

substitution

$$\sigma : V \to T(\Sigma, V)$$

with $\sigma(x) \neq x$ for only finitely many $x \in V$.

The set of all such substitutions is denoted by $\mathcal{S}ub(T(\Sigma, V))$ or simply $\mathcal{S}ub$.

$\mathcal{S}ub(T(\Sigma, V))$

Def. (3.1.6) A substition σ (on variables) is *extended* to a mapping on terms $\hat{\sigma}$: $T(\Sigma, V) \to T(\Sigma, V)$ along the term structure (i.e. inductively):

extended $\hat{\sigma}$

- $\hat{\sigma}(x) := \sigma(x)$
- $\hat{\sigma}(f(s_1 \dots s_n)) := f(\hat{\sigma}(s_1) \dots \hat{\sigma}(s_n))$

Def. (3.1.6) The *composition* $\sigma\tau$ of substitutions is defined by $\sigma\tau(x) := \hat{\sigma}(\tau(x))$.

composition $\sigma\tau$

We usually write $\sigma(t)$ for $\hat{\sigma}(t)$, since it is clear from the context what is meant.

$\sigma(t)$ for $\hat{\sigma}(t)$

Def. (3.1.6) t is an *instance* of s, written $t \gtrsim s$, if for some substitution $\sigma(s) = t$.

instance $t \gtrsim s$

Properties of Substitutions

Def. (3.1.6) The *domain* $\mathcal{D}om(\sigma)$ is the set of variables not mapped to themselves i.e. $\mathcal{D}om(\sigma) := \{x \in V | \sigma(x) \neq x\}$

domain $\mathcal{D}om(\sigma)$

For $x \in \mathcal{D}om(\sigma)$, σ *instantiates* x.

instantiates

The *range* $\mathcal{R}an(\sigma)$ is the set of result terms of σ, i.e. $\mathcal{R}an(\sigma) := \{\sigma(x) | x \in \mathrm{dom}(\sigma)\}$.

range $\mathcal{R}an(\sigma)$

The *variable range* $\mathcal{VR}an(\sigma)$ is the set of variables in σ's range, i.e. $\mathcal{VR}an(\sigma) := \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(\sigma(x))$

variable range $\mathcal{VR}an(\sigma)$

Identities and Equality

Def. (3.1.7) A Σ-*identity* is a pair of terms $(s,t) \in T(\Sigma, V)$, which we write $s \approx t$. **identity**
s is the *left-hand side*, t is the *right-hand side*. **left-hand side**
right-hand side

Reduction

Basic Reduction Step

Def. (3.1.8) For a set of identities E, *reduction* is a binary relation on terms with **reduction**

$$s[\sigma(l)]_p \to_E s[\sigma(r)]_p$$

for suitable σ, p, and $(l,r) \in E$:

An instance of the left-hand side of some equation at some position is replaced by the corresponding instance of the right-hand side of the equation.

Formally, this can be written as

$$s \to_E t \text{ iff } \exists (l,r) \in E, p \in \mathcal{P}os(s), \sigma \in \mathcal{S}ub.\ s|_p = \sigma(l)\ \wedge t = s[\sigma(r)]_p$$

Derived Reduction Relations

The following definitions are not specific to the above notion of reduction, but capture the idea of "reduction" more abstractly as a binary relation. The definitions in [1] are therefore found earlier on, in §2.1.

Def. (§2.1.1) For binary relations $R \subseteq A \times B$ and $S \subseteq B \times C$, the *composition* of relations is defined as **composition**

$$R \circ S := \{(x,z) \in A \times C | \exists y \in B.\ (x,y) \in R \text{ and } (y,z) \in S\}$$

Def. (2.1.1) For a binary relation $\to \subseteq (A \times A)$, we define the following derived relations.

basis: inductive definition

$\overset{0}{\to}$	:=	$\{(x,x) \mid x \in A\}$	**identity**
$\overset{i+1}{\to}$	:=	$\overset{i}{\to} \circ \to$	$(i+1)$**-fold composition**

derived: multi-step

$\overset{=}{\to}$	:=	$\to \cup \overset{0}{\to}$	**reflexive closure**
$\overset{+}{\to}$	:=	$\bigcup_{i>0} \overset{i}{\to}$	**transitive closure**
$\overset{*}{\to}$	:=	$\overset{+}{\to} \cup \overset{0}{\to}$	**reflexive transitive closure**

derived: symmetric / reduction with reverse equations

$\overset{-1}{\to}$	:=	$\{(y,x) \mid x \to y\}$	**inverse**
$\leftarrow$	:=	$\overset{-1}{\to}$	
$\leftrightarrow$	:=	$\to \cup \leftarrow$	**symmetric closure**
$\overset{+}{\leftrightarrow}$	:=	$(\leftrightarrow)^+$	**transitive symmetric closure**
$\overset{*}{\leftrightarrow}$	:=	$(\leftrightarrow)^*$	**reflexive transitive symmetric closure**

Termination

Notation and terminology for multi-step reductions

Def. (2.1.2)

1. x is *reducible* iff there is y with $x \to y$ — **reducible**
2. x is *in normal form* (*irreducible*) iff it is not reducible — **in normal form** **irreducible**
3. y is *a normal form of* x iff $x \xrightarrow{*} y$ and y is in normal form. — **a normal form of**
4. If x has a unique normal form, we denote it by $x\downarrow$. — $x\downarrow$
5. y is a *direct successor* of x if $x \to y$. — **direct successor**
6. y is a *successor* of x iff $x \xrightarrow{+} y$. — **successor**

Def. (2.1.3) A reduction $\to$ is

1. *terminating* iff there is *no infinite* descending chain $t_0 \to t_1 \to \cdots$ — **terminating**
2. *normalizing* every term *has a* normal form — **normalizing**

Note: "normalizing" is weaker than "terminating", because "normalizing" only requires the *existence* of a finite reduction sequence from each element, while "termininating" requires that *all* such reduction sequences are finite.

Well-founded induction

Idea: treat a terminating reduction $\to$ as a well-founded relation / order on the terms, i.e. consider $\xrightarrow{+}$ as an order $>$.

Usual well-founded induction principle for $(A, <)$: Show that $P(y)$ under the assumption that P already holds for smaller elements x.

$$\frac{\forall y \in A.\ (\forall x \in A.\ x < y \Rightarrow P(x)) \Rightarrow P(y)}{\forall x \in A.\ P(x)}$$

Transfer to terminating reductions by considering successors as "smaller" (because they require fewer steps until they reach a normal form).

$$\frac{\forall s \in T.\ (\forall t \in T.\ s \xrightarrow{+} t \Rightarrow P(t)) \Rightarrow P(s)}{\forall s \in T.\ P(s)}$$

Finitely Branching Reductions

Def. (2.2.3) A reduction $\to$ is

1. *finitely branching* if each element has finitely many *direct* successors — **finitely branching**
2. *globally finite* if each element has only finitely many successors — **globally finite**
3. *acyclic* if for no element t we have $t \xrightarrow{+} t$ — **acyclic**

Proving Termination

Lexicographic Extensions of Orders

Def. (§2.4) For two strict orders $(A, >_A)$ and $(B, >_B)$ define the *lexicographic product* $>_{A\times B}$ on $A \times B$ by **lexicographic product** $>_{A\times B}$

$$(x, y) >_{A\times B} (x', y') \quad \text{iff} \quad (x >_A x') \vee (x = x' \wedge y >_B y')$$

Def. (§2.4) By extension, the lexicographic product of n orders $(A_i, >_i)$ is given by

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

If all A_i are the same, we write $>^n_{lex}$ for this relation. $>^n_{lex}$

Def. (§2.4) By extension, we define a lexicographic order on strings of different lengths by

$$u >^*_{lex} v \quad \text{iff} \quad (|u| > |v|) \vee (|u| = |v| \wedge u >^{|u|}_{lex} v)$$

Def. (3.1.3) Given a strict order $(A, >)$, the *prefix order* $>_{prefix}$ on strings A^* is defined by **prefix order** $>_{prefix}$

$$u >_{prefix} v \quad \text{iff} \quad \exists u' \neq \varepsilon.\ u = vu'$$

Def. (§2.4) Given a strict order $(A, >)$ define $>_{Lex}$ as $>_{Lex}$

$$u >_{Lex} v \quad \text{iff} \quad u >_{prefix} v \vee (\exists wu'v'ab.u = wau' \wedge v = wbv' \wedge a > b$$

i.e. either v is a proper prefix of u or they share a common prefix w and the first symbol after w is greater in u than in v.

Note: even for terminating $>$, the order $>_{Lex}$ need not be terminating (example: lecture).

Multiset Orders

Def. (2.5.1) For set A, a *multiset* over A is a function $M\colon A \to \mathbb{N}$ **multiset**
(which for $a \in A$ gives the number of occurrences of a in M).

A multiset M is *finite* if there are finitely many $a \in A$ with $M(a) > 0$. **finite**

The set of all finite multisets over A is denoted by $\mathcal{M}(A)$.

Def. (2.5.2) Define the basic operations on multisets as follows:

$x \in M$	iff $M(x) > 0$
$M \subseteq N$	iff $\forall x \in A.M(x) \leq N(x)$
$(M \cup N)(x)$	$:= M(x) + N(x)$
$(M - N)(x)$	$:= M(x) \dot{-} N(x)$
	where $m \dot{-} n$ is $m - n$ if $m \geq n$ and is 0 otherwise

Def. (2.5.3) For given strict order $(A, >)$, define the *multiset order* $>_{mul}$ on $\mathcal{M}(A)$ as **multiset order** $>_{mul}$

$M >_{mul} N$ iff $\exists X, Y \in \mathcal{M}(A)$.

$X \neq \emptyset \wedge X \subseteq M \wedge$	a non-empty multiset from M
$N = (M - Y) \cup Y \wedge$	is removed and replaced by a multiset
$\forall y \in Y.\ \exists x \in X.\ x > y$	whose elements are bounded by those removed

Reduction- and Simplification Orders

Def. (5.2.1) For signature Σ and (countably infinite) set of variables V, a strict order $>$ on $T(\Sigma, V)$ is called a *rewrite order* iff it is

rewrite order

- *compatible with Σ-operations*, i.e. for terms s, s', t_i and n-ary symbol $f \in \Sigma$ $s > s'$ implies

  $$f(t_1 \dots t_{i-1}, s, t_{i+1} \dots t_n) > f(t_1 \dots t_{i-1}, s', i+1_n \dots i+1_)$$

  compatible with Σ-operations

- *closed under substitutions*, i.e. for terms s, t and substitutions σ, $s > t$ implies $\hat{\sigma}(s) > \hat{\sigma}(t)$.

  closed under substitutions

A *reduction order* is a well-founded rewrite order.

reduction order

Note: we could have replaced the first condition equivalently by: $>$ must be *closed under contexts*, i.e. $s > s'$ implies $t[s]_p > t[s']_p$ (for any position p in t).

closed under contexts

Note (Theorem 5.2.3) that a rewriting system is terminating iff there is a reduction order $>$ satisfying $l > r$ iff for all rewrite rules $l \to r$.

Def. (5.4.1) A strict order $>$ on $T(\Sigma, V)$ is called a *simplification order* iff it is a rewrite order and satisfies the *subterm property*: for term t and position $p \in \mathcal{P}os(t) \setminus \{\varepsilon\}$ we have $t > t|_p$.

simplification order
subterm property

Equivalently, we could have required that for n-ary function symbols f and variables $x_1 \dots x_n$, we have $f(x_1 \dots x_n) > x_i$ for all $1 \leq i \leq n$.

Note (Theorem 5.4.8): Every simplification order is a reduction order. However, the converse is not true, i.e. not all reduction orders are also simplification orders.

Simplification Orders are Reduction Orders

Def. (5.4.3) A partial order $\succeq$ on a set A is a *well-partial-order* (*wpo*) iff for every infinite sequence $a_1, a_2, a_3, \dots$ of elements of A there are indices $i < j$ with $a_i \preceq a_j$.

well-partial-order
wpo

Def. (5.4.2) The *homeomorphic embedding* is a relation on terms $T(\Sigma, X)$ with $s \trianglerighteq_{emb} t$ iff one of the following conditions hold:

homeomorphic embedding
$s \trianglerighteq_{emb} t$

1. $s = x = t$ for a variable $x \in X$
2. $s = f(s_1 \dots s_n)$, $t = f(t_1 \dots t_n)$ and $s_i \trianglerighteq_{emb} t_i$ for all $1 \leq i \leq n$
3. $s = f(s_1 \dots s_n)$ and $s_j \trianglerighteq_{emb} t$ for some $1 \leq j \leq n$

Lexicographic Path Orders

Def. (5.4.12) For a finite signature Σ and a strict order $>$ on Σ, the *lexicographic path order* $>_{lpo}$ on terms over Σ is defined by $s >_{lpo} t$ iff

lexicographic path order

(LPO1) $t \in \mathcal{V}ar(s)$ and $s \neq t$, *or*

(LPO2) $s = f(s_1 \dots s_m)$, $t = g(t_1 \dots t_n)$ *and*

- **(LPO2a)** there is i with $s_i >_{lpo} t$, *or*
- **(LPO2b)** $f > g$ and $s >_{lpo} t_j$ for all $1 \leq j \leq n$, *or*
- **(LPO2c)** $f = g$, $s >_{lpo} t_j$ for all $1 \leq j \leq n$ and there is $1 \leq i \leq m$ with $s_1 = t_1, \dots, s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$

Def. (§5.4.2, p. 121) In the setting of the previous definition, the *multiset path order* $>_{mpo}$ induced by the strict order $>$ is defined in the same way as the lexicographic path order, except that in clause (LPO2c) the lexicographic comparison of children is replaced by a multiset order:

multiset path order

$f = g$, $s >_{mpo} t_j$ for all $1 \leq j \leq n$ and
$\{s_1 \dots s_n\} >_{mul} \{t_1 \dots t_n\}$.

For the *recursive path order with status*, each function symbol is assigned a *status* that decides whether its child terms are to be compared lexicographically or as multisets.

recursive path order with status
status

Knuth-Bendix Orders

Def. (§5.4.4) For a finite signature Σ, variable set V, and strict order $>$ on Σ
a *weight function* $w\colon \Sigma \cup V \to \mathbb{R}_0^+$ is *admissible* for $>$ iff it satisfies the properties:

weight function
admissible

1. For all variables $x \in V$, $w(x) = w_0$ for some $w_0 > 0$ and for all constants $c \in \Sigma^{(0)}$ we have $w(c) \geq w_0$.
2. If f is a unary function (in $\Sigma^{(1)}$) and $w(f) = 0$, then f is the greatest element in Σ, i.e. $f \geq g$ for all $g \in \Sigma$.

Let $|t|_x$ and $|t|_f$ denote the number of occurrences of variable x and symbol f, respectively, in term t. $|t|_x$

We extend the weight function w to terms directly by considering the number of occurrences of each symbol/variable and its respective weight. $|t|_f$

$$w(t) := \sum_{x \in \mathcal{V}ar(t)} w(x) \cdot |t|_x \; + \sum_{f \in \Sigma} w(f) \cdot |t|_f$$

Def. (5.4.18) For finite signature Σ with strict order $>$ and (admissible) weight function $w\colon \Sigma \cup V \to \mathbb{R}_0^+$, the *Knuth-Bendix order* $>_{kbo}$ is defined by $s >_{kbo} t$ iff

Knuth-Bendix order

(KBO1) $|s|_x \geq |t|_x$ for all $x \in V$ and $w(s) > w(t)$ *or*

(KBO2) $|s|_x \geq |t|_x$ for all $x \in V$ and $w(s) = w(t)$, and

- **(KBO2a)** $s = f^n(x)$ and $t = x$ for some unary symbol f and variable x, *or*
- **(KBO2b)** $s = f(s_1 \dots s_m)$, $t = g(t_1 \dots t_m)$ and $f > g$, *or*
- **(KBO2c)** $s = f(s_1 \dots s_n)$ and $t = f(t_1 \dots t_n)$, and for some $1 \leq i \leq n$: $s_1 = t_1 \dots s_{i-1} = t_{i-1}$ and $s_i >_{kbo} t_i$.

Polynomial Interpretations

Def. (3.2.1) For a given signature Σ, a Σ-*algebra* consists of a *carrier set* A and a mapping from function symbols $f \in \Sigma^{(n)}$ to functions $f^{\mathcal{A}} \colon A^n \to A$ (for all $n \geq 0$). **Σ-algebra** **carrier set**

Def. (5.3.1) For non-empty Σ-algebra $\mathcal{A}$ and well-founded strict order $>$ on the carrier set A, the order $>_{\mathcal{A}}$ on terms is defined by

$$s >_{\mathcal{A}} t \qquad \text{iff} \qquad \pi(s) > \pi(t) \text{ for all homomorphims } \pi$$

Note: the homomorphism is determined by it result for the variables, i.e. we are really working with *valuations* $\pi : V \to A$ that are extended to terms in the obvious way: **valuations**

$$\hat{\pi}(x) = \pi(x)$$
$$\hat{\pi}(f(s_1 \dots s_n)) = f(\hat{\pi}(s_1) \dots \hat{\pi}(s_n))$$

Def. (5.3.2) A function $F : A^n \to A$ is called *monotone* (w.r.t. a strict order $>$ on A) iff for all $1 \leq i \leq n$ **monotone**

$$a > b \quad \text{implies} \quad F(a_1 \dots a_{i-1}, a, a_{i+1} \dots a_n) > F(a_1 \dots a_{i-1}, b, a_{i+1} \dots a_n)$$

Def. (5.3.4) A *polynomial interpretation* of Σ is a Σ-algebra $\mathcal{A}$ for which **polynomial interpretation**

1. The carrier set is $A \subseteq \mathbb{N} - \{0\}$
2. Each function symbol $f \in \Sigma^{(n)}$ is mapped to a polynomial $P_f(X_1 \dots X_n) \in \mathbb{N}[X_1 \dots X_N]$. The interpretation $f^{\mathcal{A}}$ is given by the evaluation of P_f, i.e. $f^{\mathcal{A}}(a_1 \dots a_n) := P_f(a_1 \dots a_n)$

Def. (5.3.6) A polynomial $P(X_1 \dots X_n) \in \mathbb{N}[X_1 \dots X_n]$ is a *monotone polynomial* iff it depends on all its indeterminates (i.e. each indeterminate occurs in at least one monomial (with a non-zero exponent)). **monotone polynomial**

A *monotone polynomial interpretation* is a polynomial interpretation where all function symbols are associated with monotone polynomials. **monotone polynomial interpretation**

Def. (§5.3, p. 107) The reduction order induced by a monotone polynomial interpretation is called a *polynomial order*. **polynomial order**

Confluence

Basic Definitions

Def. (§2.1.1) x and y are *joinable*, written $x \downarrow y$, iff there is a z such that $x \xrightarrow{*} z \xleftarrow{*} y$.

joinable

Def. (2.1.3) A reduction $\to$ is called

Church Rosser	iff	$x \overset{*}{\leftrightarrow} y \Rightarrow x \downarrow y$
confluent	iff	$y_1 \xleftarrow{*} x \xrightarrow{*} y_2 \Rightarrow y_1 \downarrow y_2$
convergent	iff	it is both confluent and terminating

Church Rosser
confluent
convergent

Def. (2.1.4) A reduction $\to$ is *semi-confluent* iff $y_1 \leftarrow x \xrightarrow{*} y \Rightarrow y_1 \downarrow y_2$

semi-confluent

Def. (2.7.1) A reduction $\to$ is *locally confluent* iff $y_1 \leftarrow x \to y_2 \Rightarrow y_1 \downarrow y_2$.

locally confluent

Syntactic Unification

Def. (4.5.1) A substitution σ is *more general* than substitution σ', written $\sigma \lesssim \sigma'$, if $\sigma' = \delta\sigma$ for some δ. Then σ' is an *instance* of σ.

Write $\sigma \sim \sigma'$ if $\sigma \lesssim \sigma'$ and $\sigma' \lesssim \sigma$.

more general
$\lesssim$
instance
$\sim$

Def. (§4.5) An substitution σ is a *renaming* if σ is injective and $\mathcal{R}an(\sigma) \subseteq V$.

renaming

Def. (4.5.4) A *unification problem* is a set of equations $S = \{s_1 =^? t_1 \dots s_n =^? t_n$.
A substitution σ is a *unifier* or *solution* of S if $\sigma s_i = \sigma t_i$ for all $i = 1, \dots, n$.

The set of unifiers of S is denoted by $\mathcal{U}(S)$.

σ is a *most general unifier* (*mgu*) of S if $\sigma \in \mathcal{U}(S)$ and $\sigma \lesssim \sigma'$ for all $\sigma' \in \mathcal{U}(S)$.

unification problem
unifier
solution
$\mathcal{U}(S)$
most general unifier
mgu

Critical Pairs

Def. (3.1.3) Define the *prefix order* on positions $p \leq q$ iff as an integer-string, p is a prefix of q
i.e. there exists p' such that $q = pp'$

p is *above* q if $p \leq q$. Analogously: *below*.

By extension: *strictly above* for $p < q$ and *strictly below*.

Positions p, q are *parallel* (written $p \,||\, q$) if they are incomparable by $\leq$

prefix order
$p \leq q$
above
below
strictly above
strictly below
parallel
$p \,||\, q$

Def. (6.2.1) Let $l_i \to r_i$, $i = 1, 2$ be two rules with disjoint variables ($\mathcal{V}ar(l_1, r_1) \cap \mathcal{V}ar(l_2, r_2) = \emptyset$, if necessary by renaming).

Then let $p \in \mathcal{P}os(l_1)$ with $l_1|_p$ not a variable. Let θ be an mgu of $l_1|_p =^? l_2$. In this case, we say that the two rules *overlap*.

The *critical pair* given by this overlap is $(\theta r_2, (\theta l_1)[\theta r_2]_p)$.

The critical pairs of between any two rules of a TRS R are denoted by $CP(R)$. Overlaps with renamed variants of rules with themselves are included, except at the root position.

overlap
critical pair

References

[1] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, Cambridge, 1998.