Contents

1 Introduction **2**

2 Equational Unification **2**

3 Boolean Rings **6**

3.1 Polynomials 7
3.2 Unification 9
3.3 Successive variable elimination 11
3.4 Löwenheim's formula 13
3.5 Conclusion 16

1 Introduction

We already know syntactic unification where we try to find a substitution σ to make two terms syntactically equal. But what if we know something about the function symbols in our terms for example we know that f is idempotent (i.e. $f(x,x) \approx x$). The syntactic unification problem $S := \left\{ f(a,x) \stackrel{?}{=} x \right\}$ where a is a constant and x a Variable does not have a solution. This is where equational unification comes in handy because it considers equality modulo an equational theory $\approx_E$: given terms s and t is there a substitution σ such that $\sigma(s) \approx_E \sigma(t)$. Let $I \models f(x,x) \approx x$ now the substitution $\sigma := \{x \mapsto a\}$ is a solution of the I-unification problem $S := \left\{ f(a,x) \stackrel{?}{\approx}_I x \right\}$ since $f(a,a) \approx_I a$.

Here we will only consider boolean unification in more detail but there are also other interesting equational theories like AC the theory that axiomatizes associativity and commutativity [1].

2 Equational Unification

In the following let E be a set of identities of the form $\{e_1 \approx f_1, \ldots, e_n \approx f_n\}$. Furthermore let $\mathcal{S}ig(E)$ denote the set of all function symbols occurring in E. Let Σ be a finite set of function symbols and a superset of $\mathcal{S}ig(E)$.

Definition 2.1. An **E-unification problem** over Σ is a finite set S of the form $S = \left\{ s_1 \stackrel{?}{\approx}_E t_1, \ldots, s_n \stackrel{?}{\approx}_E t_n \right\}$ with $s_1, \ldots, s_n, t_1, \ldots, t_n \in T(\Sigma, V)$, V being a countable set of Variables.

A substitution σ is an **E-unifier** of S iff $\sigma(s_i) \approx_E \sigma(t_i)$ for all $1 \leq i \leq n$. The set of all E-unifiers of S is denoted by $\mathcal{U}_E(S)$. S is **E-unifiable** iff $\mathcal{U}_E(S) \neq \emptyset$.

If an E-unification problem S contains only one equation $s \stackrel{?}{\approx}_E t$ we often omit the brackets and write $S = s \stackrel{?}{\approx}_E t$. We can now classify unification problems according to the symbols occurring in them.

Definition 2.2. Let S be an E-unification problem over Σ.

- S is an **elementary** E-unification problem iff $\mathcal{S}ig(E) = \Sigma$.
- S is an E-unification problem **with constants** iff $\Sigma - \mathcal{S}ig(E) \subseteq \Sigma^{(0)}$ and $\mathcal{S}ig(E) \subset \Sigma$.
- S is an **general** E-unification problem iff $\Sigma - \mathcal{S}ig(E)$ contains an at least unary function symbol.

One *most general unifier* does not always suffice to represent $\mathcal{U}_E(S)$. In this case we need a *E-minimal E-complete set of E-unifiers* but to define this set we first need an order on substitutions.

Definition 2.3. Let X be a set of variables. A substitution σ is **more general** modulo $\approx_E$ than a substitution σ' on X iff there is a substitution δ such that $\delta(\sigma(x)) \approx_E \sigma'(x)$ for all $x \in X$. We denote this by $\sigma \lesssim_E^X \sigma'$.

$\lesssim_E^X$ is a is a quasi order since it obviously is reflexive and transitive. To see why we only demand equality modulo $\approx_E$ on X and not on all Variables like we did in syntactic unification we fist need to define *E-minimal E-complete sets*.

Definition 2.4. Let S be an E-unification problem over Σ and let $X := \mathcal{V}ar(S)$. An E**-complete** set of S is a set of substitutions $\mathcal{C}$ that satisfies the following properties.

- each $\sigma \in \mathcal{C}$ is an E-unifier of S
- for all $\theta \in \mathcal{U}_E(S)$ there exists a $\sigma \in \mathcal{C}$ such that $\sigma \lesssim_E^X \theta$

An E**-minimal** E**-complete** set is an E-complete set $\mathcal{M}$ that satisfies the additional property

- for all $\sigma, \sigma' \in \mathcal{M},\ \sigma \lesssim_E^X \sigma'$ implies $\sigma = \sigma'$.

The substitution σ is a **most general** E**-unifier** (mgu) of S iff $\{\sigma\}$ is an E-minimal E-complete set of S.

Now to understand why the restriction to X makes sense note that by the restriction to Variables in X more substitutions are comparable with respect to $\lesssim_E^X$ since we do not demand equality modulo $\approx_E$ on all Variables. Lets denote the Variables occurring in an E-unification problem S by $\mathcal{V}ar(S)$. It is easy to see that if $X = \mathcal{V}ar(S)$, σ is an E-unifier of S and $\sigma \lesssim_E^X \sigma'$ then σ' is also an E-unifier of S. This only shows that restriction to X does not do any damage but the reason it is useful is that there are E-unification problems S for which any E-minimal E-complete set of E-unifiers has to contain Variables not occurring in S.

Lets consider a small example, let $\sigma := \{x \mapsto f(y)\}$ be in $\mathcal{M}$ an E-minimal E-complete set of E-unifiers of S with $\mathcal{V}ar(S) = \{x\}$ and $a \not\approx_E x$. Clearly $\sigma' := \{x \mapsto f(a)\}$ is also an E-unifier of S but σ and σ' are incomparable with respect to $\lesssim_E^{\{x,y\}}$. The substitution $\delta := \{y \mapsto a\}$ does not work here since $\delta(\sigma(y)) = a \not\approx_E y = \sigma'(y)$ which means there has to be another unifier σ'' in $\mathcal{M}$ with $\sigma'' \lesssim_E^{\{x,y\}} \sigma$. But if we restrict X to $\{x\}$ we only need that $\delta(\sigma(x)) = f(a) = \sigma'(x)$, so $\sigma \lesssim_E^{\{x\}} \sigma'$ holds. We see that E-minimal E-complete sets can become unnecessary large if we consider all Variables.

Now let us consider an example in which an E-minimal E-complete set contains infinitely many elements. Let $A := \{x + (y + z) \approx (x + y) + z\}$ be a set of identities and $S := \left\{x + a \stackrel{?}{\approx}_A a + x\right\}$ an A-unification problem over $\Sigma := \{+, a\}$. For $n > 0$, we define substitutions σ_n inductively as follows:

$$\sigma_1 := \{x \mapsto a\}$$
$$\sigma_{n+1} := \{x \mapsto a + \sigma_n(x)\}$$

Since A axiomatizes associativity we can omit the brackets and give an explicit definition of σ_n.

$$\sigma_n := \{x \mapsto \underbrace{a + \cdots + a}_{n \times a}\}$$

Now it is easy to see that all σ_n are A-unifiers of S. Lets consider an arbitrary A-unifier θ of S. The substitution θ has the form $\theta(x) := x_1 + \cdots + x_n$ where the $x_i's$ are either a or a variable. Since θ is an A-unifier of S we know that:

$$\begin{array}{lll}
 & & \theta(x) + a \approx_A a + \theta(x) \\
\iff & & x_1 + \cdots + x_n + a \approx_A a + x_1 + \cdots + x_n \\
\implies & x_1 = x_n = a & a + x_2 + \cdots + a + a \approx_A a + a + \cdots + x_{n-1} + a \\
\implies & x_2 = x_{n-1} = a & a + a + \cdots + a + a + a \approx_A a + a + a + \cdots + a + a \\
\vdots & & \vdots \\
\implies & & \underbrace{a + \cdots + a}_{n+1 \times a} \approx_A \underbrace{a + \cdots + a}_{n+1 \times a}
\end{array}$$

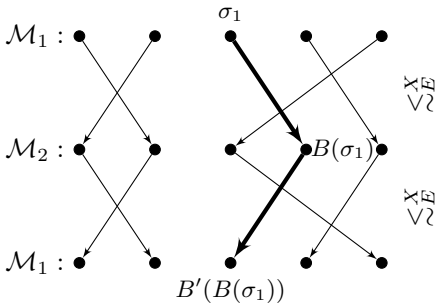
So $\theta(x) \approx_A \sigma_n(x)$ which implies $\sigma \lesssim_A^{\{x\}} \theta$. Since we picked θ arbitrarily this yields A-completeness of the set $\mathcal{M} := \bigcup_{n>0} \{\sigma_n\}$. All σ_n are distinct and map x to ground terms. Hence they are pairwise incomparable with respect to $\lesssim_A^{\{x\}}$. This yields A-minimality of $\mathcal{M}$. We see that E-minimal E-complete sets do not need to have finite cardinality.

We denote the equivalence class induced by $\lesssim_E^X$ with $\sim_E^X$.

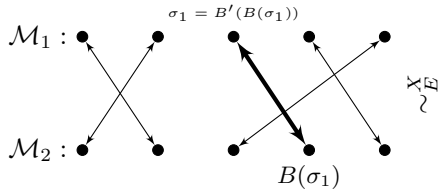
$$\sigma \sim_E^X \sigma' \text{ iff } \sigma \lesssim_E^X \sigma' \text{ and } \sigma' \lesssim_E^X \sigma$$

Lemma 2.5. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be E-minimal E-complete sets of S. Then there exists a bijective mapping $B : \mathcal{M}_1 \mapsto \mathcal{M}_2$ such that $\sigma_1 \sim_E^X B(\sigma_1)$ for all $\sigma_1 \in \mathcal{M}_1$.*

Proof. We define a mapping $B : \mathcal{M}_1 \mapsto \mathcal{M}_2$ such that $B(\sigma_1) \lesssim_E^X \sigma_1$ for all $\sigma_1 \in \mathcal{M}_1$. This is possible since $\mathcal{M}_1 \subseteq \mathcal{U}_E(S)$ and E-completeness of $\mathcal{M}_2$ yields that for every $\sigma_1 \in \mathcal{M}_1$ there exists a $\sigma_2 \in \mathcal{M}_2$ such that $\sigma_2 \lesssim_E^X \sigma_1$. We define $B' : \mathcal{M}_2 \mapsto \mathcal{M}_1$ in a similar way.



Since by definition $B'(B(\sigma_1)) \lesssim_E^X B(\sigma_1) \lesssim_E^X \sigma_1$ E-minimality of $\mathcal{M}_1$ implies that $B'(B(\sigma_1)) = \sigma_1$ for all $\sigma_1 \in \mathcal{M}_1$. Symmetrically, $B(B'(\sigma_2)) = \sigma_2$ for all $\sigma_2 \in \mathcal{M}_2$. It follows that B is a bijection and $B' = B^{-1}$.



□

The most interesting consequence of this Lemma is that E-minimal E-complete sets of the same S always have the same cardinality. This allows us to classify equational theories $\approx_E$ by the existence and possible cardinalities of E-minimal E-complete sets of E-unification problems.

Definition 2.6. The equational theory $\approx_E$ is of **unification type**

unitary iff for all E-unification problems S there exists an E-minimal E-complete set of cardinality ≤ 1.

finitary iff for all E-unification problems S there exists an E-minimal E-complete set with finite cardinality.

infinitary iff for all E-unification problems S there exists an E-minimal E-complete set, and there exists an E-unification problem for which this set is infinite.

zero iff there exists an E-unification problem that does not have an E-minimal E-complete set.

Note that if the E-unification problem S has no E-unifiers then the empty set is an E-minimal E-complete set of S. The empty set is E-complete because there are no $\sigma \in \emptyset$ and $\mathcal{U}_E$ is empty. E-minimality holds trivially. This is the reason we allow the cardinalities 0 and 1 in the *unitary* case. An example for a $finitary$ theory that is not *unitary* is $\mathcal{C} := \{f(x,y) \approx f(y,x)\}$ which axiomatizes commutativity. With $\mathcal{A} := \{x + (y + z) \approx (x + y) + z\}$ the theory that axiomatizes associativity we have already seen an example for an $infinitary$ equational theory. In the definition of the $unification\ types$ we allowed for arbitrary E-unification problems but if we distinguish between elementary E-unification problems, E-unification problems with constants and general E-unification problems we might end up with different $unification\ types$.

3 Boolean Rings

$$B := \left\{ \begin{array}{cc} x + y \approx y + x, & x * y \approx y * x, \\ (x + y) + z \approx x + (y + z), & (x * y) * z \approx x * (y * z), \\ x + x \approx 0, & x * x \approx x, \\ 0 + x \approx x, & 0 * x \approx 0, \\ x * (y + z) \approx (x * y) + (x * z), & 1 * x \approx x \end{array} \right\}$$

Since $+$ and $*$ are associative we can omit most of the brackets. Furthermore we often write xy instead of $x * y$. Lets consider an interpretation of B the two element boolean ring $\mathcal{B}_2$ with the carrier set $\mathbf{2} := \{\bot, \top\}$ where $*$ is "and" and $+$ is "exclusive or":

$$\begin{array}{cc} (x + y)^{\mathcal{B}_2} := \left(x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2}\right) \vee \left(\neg x^{\mathcal{B}_2} \wedge y^{\mathcal{B}_2}\right) & (x * y)^{\mathcal{B}_2} := x^{\mathcal{B}_2} \wedge y^{\mathcal{B}_2} \\ 0^{\mathcal{B}_2} := \bot & 1^{\mathcal{B}_2} := \top \end{array}$$

It is easy to see that $\mathcal{B}_2$ is indeed a model of B. Furthermore we can transform a term back from an Boolean algebra into Boolean ring theory:

$$\begin{aligned} x \wedge y &\mapsto x * y \\ x \vee y &\mapsto x + y + x * y \\ \neg x &\mapsto x + 1 \end{aligned}$$

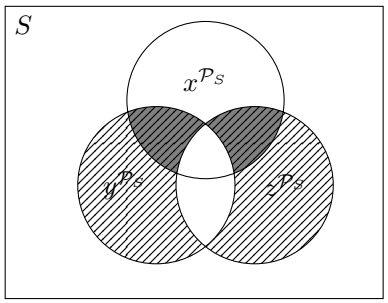
We work with + and * instead of $\vee, \wedge$ and $\neg$ because in Boolean rings we have a very convenient normal form which makes the following proofs easier.
Lets consider another model of B the powerset interpretation $\mathcal{P}_S$ with the carrier set 2^S:

$$\begin{array}{cc} (x + y)^{\mathcal{P}_S} := x^{\mathcal{P}_S} \Delta y^{\mathcal{P}_S} & (x * y)^{\mathcal{P}_S} := x^{\mathcal{P}_S} \cap y^{\mathcal{P}_S} \\ 0^{\mathcal{P}_S} := \emptyset & 1^{\mathcal{P}_S} := S \end{array}$$

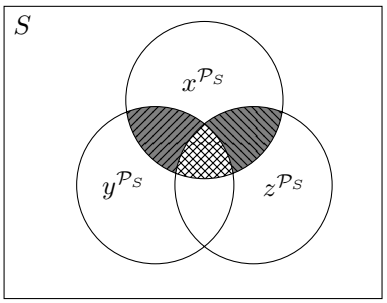
Where $x \Delta y := (x \setminus y) \cup (y \setminus x)$ is the symmetric difference of x and y. It is easy to see why $\mathcal{P}_S$ is a model of B. Lets just consider distributivity in detail.

$$\begin{aligned} (x * (y + z))^{\mathcal{P}_S} &= x^{\mathcal{P}_S} \cap \left(y^{\mathcal{P}_S} \Delta z^{\mathcal{P}_S}\right) \\ &= x^{\mathcal{P}_S} \cap \left(\left(y^{\mathcal{P}_S} \setminus z^{\mathcal{P}_S}\right) \cup \left(z^{\mathcal{P}_S} \setminus y^{\mathcal{P}_S}\right)\right) \\ &= \left(\left(x^{\mathcal{P}_S} \cap y^{\mathcal{P}_S}\right) \setminus z^{\mathcal{P}_S}\right) \cup \left(\left(x^{\mathcal{P}_S} \cap z^{\mathcal{P}_S}\right) \setminus y^{\mathcal{P}_S}\right) \\ &= \left(\left(x^{\mathcal{P}_S} \cap y^{\mathcal{P}_S}\right) \setminus \left(x^{\mathcal{P}_S} \cap z^{\mathcal{P}_S}\right)\right) \cup \left(\left(x^{\mathcal{P}_S} \cap z^{\mathcal{P}_S}\right) \setminus \left(x^{\mathcal{P}_S} \cap y^{\mathcal{P}_S}\right)\right) \\ &= \left(x^{\mathcal{P}_S} \cap y^{\mathcal{P}_S}\right) \Delta \left(x^{\mathcal{P}_S} \cap z^{\mathcal{P}_S}\right) \\ &= ((x * y) + (x * z))^{\mathcal{P}_S} \end{aligned}$$

Here is a less formal explanation for why this identity holds in $\mathcal{P}_S$.



$(x * (y + z))^{\mathcal{P}_S}$



$((x * y) + (x * z))^{\mathcal{P}_S}$

This small example should just show that there are other models of B with rather common interpretations of $+$ and $*$ apart from $\mathcal{B}_2$. Note that if $|S| = 1$ then $\mathcal{P}_S$ and $\mathcal{B}_2$ are isomorphic. In the following we will only consider elementary B-unification problems since additional function symbols would just complicate the following proofs.

3.1 Polynomials

Definition 3.1. A product of distinct variables is a **monomial** (e.g. xyz). And a sum of distinct monomials is a **polynomial** (e.g. $x + xy + yz$).

We compare monomials and polynomials modulo commutativity and associativity. So two monomials are distinct iff the sets of variables occurring in them are distinct and two polynomials are distinct iff the sets of their monomials are distinct. Here are some examples for clarification:

$$\begin{aligned} yxz &= zyx & \qquad xy + yz &= zy + xy \\ yx &\neq yxz & \qquad xy + yz &\neq xy \end{aligned}$$

Note that we did not introduce a new symbol for equality of polynomials and just use the same as for syntactic equality since it will be clear from the context which one we mean. Now we can transform every term over $\{0, 1, +, *\}$ into a (w.r.t. equality of polynomials) unique $\approx_B$-equivalent polynomial, its **polynomial form**. Since 1 is the neutral element of $*$ we write 1 for the polynomial containing only the empty monomial correspondingly we identify 0 with the empty polynomial. Now the polynomial form can be computed recursively as follows:

$x, 0, 1$: This is the base case, variables and the constants 0 and 1 are already polynomials.

$t_1 + t_2$: Let p_1 and p_2 be the polynomial forms of t_1 and t_2 the polynomial form of $t_1 + t_2$ is obtained by removing all pairs of equivalent monomials from $p_1 + p_2$. Since we have $\{0 + x \approx x, x + x \approx 0\} \subseteq B$ this rule preserves $\approx_B$-equivalence.

$t_1 * t_2$: Let $p_1 = m_1 + \cdots + m_k$ and $p_2 = n_1 + \cdots + n_l$ be the polynomial forms of t_1 and t_2. The polynomial form of $t_1 * t_2$ is obtained by removing all pairs of equivalent monomials from $p_1 * p_2$ which when multiplied out is the sum

$$m_1 * n_1 + \cdots + m_1 * n_l + \cdots + m_k * n_1 + \cdots + m_k * n_l$$

where the product of two monomials $m = x_1 \dots x_r$ and $n = y_1 \dots y_s$ is the monomial obtained by removing repeated occurrences of the same variable from $x_1 \dots x_r y_1 \dots y_s$. Since we have $\{x * x \approx x\} \in B$ this rule preserves $\approx_E$-equivalence. Note that if $t_1 = 1$ then we multiply every monomial in p_2 with the empty monomial which does not change anything so the result is as expected just p_2.

The polynomial form of t is denoted by $t{\downarrow_P}$. It is easy to see that all rules preserve $\approx_B$-equivalence and hence $t \approx_B t{\downarrow_P}$.

Theorem 3.2. *The following statements are equivalent:*

1. $\mathcal{B}_2 \models s \approx t$,
2. $s{\downarrow_P} = t{\downarrow_P}$,
3. $s \approx_B t$.

We will not proof this theorem but instead look at a bigger example. Let $s := (y+1) * (x+y) + (y+1) * x$ and $t := 0$ and go through 1, 2 and 3 from theorem 3.2.

1. $\mathcal{B}_2$ is a model of $s \approx t$.

$$\begin{aligned}
s^{\mathcal{B}_2} &= \left((y+1)\,(x+y) + (y+1)\,x\right)^{\mathcal{B}_2} \\
&= \left(s_1^{\mathcal{B}_2} \wedge \neg s_2^{\mathcal{B}_2}\right) \vee \left(\neg s_1^{\mathcal{B}_2} \wedge s_2^{\mathcal{B}_2}\right) \\
&\quad s_1 = (y+1)^{\mathcal{B}_2} \wedge (x+y)^{\mathcal{B}_2} \qquad\qquad s_2 = (y+1)^{\mathcal{B}_2} \wedge x^{\mathcal{B}_2} \\
&\qquad = \neg y^{\mathcal{B}_2} \wedge (x+y)^{\mathcal{B}_2} \qquad\qquad = \neg y^{\mathcal{B}_2} \wedge x^{\mathcal{B}_2} \\
&\qquad = \neg y^{\mathcal{B}_2} \wedge \left(\left(x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2}\right) \vee \left(\neg x^{\mathcal{B}_2} \wedge y^{\mathcal{B}_2}\right)\right) \\
&\qquad = x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2} \\
s^{\mathcal{B}_2} &= \left(\left(x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2}\right) \wedge \neg \left(\neg y^{\mathcal{B}_2} \wedge x^{\mathcal{B}_2}\right)\right) \vee \left(\neg \left(x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2}\right) \wedge \left(\neg y^{\mathcal{B}_2} \wedge x^{\mathcal{B}_2}\right)\right) \\
&= \left(x^{\mathcal{B}_2} \wedge \left(\neg y^{\mathcal{B}_2} \wedge \left(y^{\mathcal{B}_2} \vee \neg x^{\mathcal{B}_2}\right)\right)\right) \vee \left(\left(\left(\neg x^{\mathcal{B}_2} \vee y^{\mathcal{B}_2}\right) \wedge \neg y^{\mathcal{B}_2}\right) \wedge x^{\mathcal{B}_2}\right) \\
&= \left(x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2} \wedge \neg x^{\mathcal{B}_2}\right) \vee \left(\neg x^{\mathcal{B}_2} \wedge \neg y^{\mathcal{B}_2} \wedge x^{\mathcal{B}_2}\right) \\
&= \bot \vee \bot \\
&= t^{\mathcal{B}_2}
\end{aligned}$$

2. The polynomial forms $s{\downarrow_P}$ and $t{\downarrow_P}$ are equal.

$$\begin{aligned} s &\approx_B (y+1)(x+y)+(y+1)x \\ &\approx_B yx+yy+x+y+yx+x \\ &\approx_B yx+yx+y+y+x+x \\ &\approx_B 0 \end{aligned}$$

$$s{\downarrow_P} = 0 = t{\downarrow_P}$$

3. $s \approx_B t$ is a consequence of B.

$$\begin{aligned} s = \quad & (y+1)(x+y)+(y+1)x \\ &\approx_B (y+1)(x+y+x) \\ &\approx_B y+y \\ &\approx_B 0 \\ = \quad & t \end{aligned}$$

A nice consequence from theorem 3.2 is that $\approx_B$ is decidable, because we could just compare the computable polynomial forms, or test semantic equality in $\mathcal{B}_2$.

3.2 Unification

Note that until now we have not said anything about unification we just introduced the equational theory B, the semantic interpretation $\mathcal{B}_2$ and showed some properties of the word problem in $\approx_B$. Now we will look at some ways to find most general B-unifiers of B-unification problems. But fist we show that unification modulo $\approx_B$ is closely related to equation solving in $\mathcal{B}_2$.

Lemma 3.3.

1. *Every solution of $s \overset{?}{\approx}_B t$ in $\mathcal{B}_2$ can be viewed as a B-unifier.*
2. *If $s \overset{?}{\approx}_B t$ has a B-unifier then $s \overset{?}{\approx} t$ has a solution in $\mathcal{B}_2$.*

Proof.

(1) Let $\varphi : V \mapsto \mathbf{2}$ be a solution of $s \overset{?}{\approx} t$ in $\mathcal{B}_2$ and $\widehat{\varphi}$ the homeomorphic extension of φ, i.e. $\widehat{\varphi}(s) = \widehat{\varphi}(t)$. From φ we now can define a mapping $\phi : V \mapsto T(\Sigma, V)$ to ground terms in the following way:

$$\phi(x) := \begin{cases} 1 & \text{if } \varphi(x) = \top \\ 0 & \text{if } \varphi(x) = \bot \end{cases}$$

for all $x \in V$. Lets denote the homeomorphic extension of ϕ by $\widehat{\phi}$. Let $\psi : \mathcal{T}(\Sigma, V) \mapsto \mathcal{B}_2$ be an arbitrary homomorphism since $\widehat{\phi}$ is a mapping to ground terms $\psi\widehat{\phi} = \widehat{\varphi}$. So the following

$$\psi\left(\widehat{\phi}(s)\right) = \widehat{\varphi}(s) = \widehat{\varphi}(t) = \psi\left(\widehat{\phi}(t)\right)$$

holds for all homomorphisms ψ and hence $\mathcal{B}_2 \models \widehat{\phi}(s) \approx \widehat{\phi}(t)$. Theorem 3.2 yields $\widehat{\phi}(s) \approx_B \widehat{\phi}(t)$, i.e. the mapping ϕ (which we got directly from the solution φ), when viewed as a substitution, is a B-unifier.

(2) Let σ be a B-unifier of $s \stackrel{?}{\approx}_B t$, i.e. $\sigma(s) \approx_B \sigma(t)$. Now theorem 3.2 yields $\mathcal{B}_2 \models \sigma(s) \approx \sigma(t)$ and hence $\psi(\sigma(s)) = \psi(\sigma(t))$ for all homomorphisms $\psi : \mathcal{T}(\Sigma, V) \mapsto \mathcal{B}_2$. Hence $\psi\sigma : V \mapsto \mathbf{2}$ is a solution in $\mathcal{B}_2$.

□

Lets consider the B-unification problem $x + y + z \stackrel{?}{\approx} z + 1$ as small example for clarification.

(1) The mapping

$$\varphi(w) := \begin{cases} \bot & \text{if } w = x \\ \top & \text{if } w \neq x \end{cases}$$

is a solution of our problem. The substitution $\sigma := \{x \mapsto 0, y \mapsto 1, z \mapsto 1\}$ is φ viewed as a B-unifier.

(2) The substitution $\sigma' := \{y \mapsto x + 1, z \mapsto 1\}$ obviously is a B-unifier. Let φ be a mapping with $\varphi(x) := \top$ for all $x \in V$ and $\widehat{\varphi}$ its homeomorphic extension.

$$\widehat{\varphi}(\sigma'(w)) = \begin{cases} (\top + 1)^{\mathcal{B}_2} = \bot & \text{if } w = y \\ 1^{\mathcal{B}_2} = \top & \text{if } w = z \\ \top & \text{otherwise} \end{cases}$$

is a solution in $\mathcal{B}_2$.

Note that $\approx_B$ is *unitary* for elementary unification and unification with constants hence we only consider elementary unification here it suffices to find an mgu and not B-minimal B-complete sets. Furthermore we can transform every B-unification problem into a B-unification problem of the form $t \stackrel{?}{\approx}_B 0$ in the following way. Let $S = \left\{ t_1 \stackrel{?}{\approx}_B s_1, \ldots, t_n \stackrel{?}{\approx}_B s_n \right\}$ be an arbitrary B-unification problem. For all $1 \leq i \leq n$ we transform $t_i \stackrel{?}{\approx}_B s_i$ into $t_i + s_i \approx_B 0$. Now we put the equation together by multiplying them in the following way

$$\begin{aligned} & (t_1 + s_1 + 1) * \cdots * (t_n + s_n + 1) \stackrel{?}{\approx}_B 1 \\ \iff \qquad & (t_1 + s_1 + 1) * \cdots * (t_n + s_n + 1) + 1 \stackrel{?}{\approx}_B 0 \end{aligned}$$

It is easy to see that our transformation does not change the set of B-unifiers. Now we will look at two ways to find mgu's of B-unification problems of the form $t \stackrel{?}{\approx}_B 0$.

3.3 Successive variable elimination

The idea of this unification algorithm is pretty simple we just eliminate variables until we have a ground B-unification problem if it has a solution then we can read off an mgu for the original problem otherwise the original problem is not B-unifiable. The following observation gives us a way to eliminate variables.

Every term t is B-equivalent to a term of the form $x * r + (x + 1) * s$ such that $x \notin \mathcal{V}ar(r, s) \subset \mathcal{V}ar(t)$. For example, we can split the polynomial form of t into two sets of monomials, those who contain x (l_x) and those who do not contain x (s). Now l is obtained by removing every occurrence of x in l_x and $r := l + s$. To see why this works consider the small example $t := yx + z$. Now $r := y + z$ and $s := z$.

$$\begin{aligned} x * (y + z) + (x + 1) * z &\approx_B xy + xz + xz + z \\ &\approx_B xy + z \\ &\approx_B t \end{aligned}$$

This simple observation is the basis of successive variable elimination. Before we come to the main theorem of successive variable elimination we introduce a strong type of most general B-unifiers the reproductive B-unifiers.

Definition 3.4. An E-unifier σ of an E-unification problem S is a **reproductive E-unifier** iff $\tau(\sigma(x)) \approx_E \tau(x)$ for every unifier τ of S and every x.

Note that for a normal mgu σ of an E-unification problem S it only has to hold that for every E-unifier τ of S there has to exist a substitution θ such that $\theta(\sigma(x)) \approx_E \tau(x)$ for every $x \in X$.

Theorem 3.5. *Let $t \approx_B x * r + (x + 1) * s$ such that $x \notin \mathcal{V}ar(r, s) \subset \mathcal{V}ar(t)$ and define $t' := r * s$.*

1. *Every B-unifier of $t \stackrel{?}{\approx}_B 0$ is a B-unifier of $t' \stackrel{?}{\approx}_B 0$.*
2. *If σ is a reproductive B-unifier of $t' \stackrel{?}{\approx}_B 0$ and $x \notin \mathcal{D}om(\sigma)$, then*

$$\sigma' := \sigma \cup \{x \mapsto x * (\sigma(r) + \sigma(s) + 1) + \sigma(s)\}$$

is a reproductive B-unifier of $t \stackrel{?}{\approx}_B 0$.

Proof.

(1) Let τ be a B-unifier of $t \stackrel{?}{\approx}_B 0$ and hence $\tau(t) \approx_B 0$.

$$\begin{aligned}
& \tau(x) * \tau(r) + (\tau(x) + 1) * \tau(s) \approx_B 0 \\
\iff \quad & \tau(x) * \tau(r) * \tau(s) + (\tau(x) + 1) * \tau(s) * \tau(r) \approx_B 0 * \tau(s) * \tau(r) \\
\iff \quad & \tau(x) * \tau(r * s) + (\tau(x) + 1) * \tau(s * r) \approx_B 0 \\
\iff \quad & (\tau(x) + \tau(x) + 1) * \tau(s * r) \approx_B 0 \\
\iff \quad & \tau(s * r) \approx_B 0
\end{aligned}$$

So τ is also a B-unifier of $t' \stackrel{?}{\approx}_B 0$.

(2) Let σ be a reproductive B-unifier of $t' \stackrel{?}{\approx}_B 0$ and $x \notin \mathcal{D}om(\sigma)$. It is easy to show that σ' is a B-unifier of $t \stackrel{?}{\approx}_B 0$:

$$\begin{aligned}
\sigma'(t) &\approx_B \sigma'(x) * \sigma'(r) + (\sigma'(x) + 1) * \sigma'(s) \\
&= (x * (\sigma(r) + \sigma(s) + 1) + \sigma(s)) * \sigma(r) + (\sigma'(x) + 1) * \sigma(s) \\
&\approx_B (x * (\sigma(r) + \sigma(s) * \sigma(r) + \sigma(r)) + \sigma(s) * \sigma(r)) + (\sigma'(x) + 1) * \sigma(s) \\
&\approx_B (x * (0 + \sigma(s * r)) + \sigma(s * r)) + (\sigma'(x) + 1) * \sigma(s) \\
&\approx_B (x * \sigma(t') + \sigma(t')) + (\sigma'(x) + 1) * \sigma(s) \\
&\approx_B 0 + (x * (\sigma(r) + \sigma(s) + 1) + \sigma(s) + 1) * \sigma(s) \\
&\approx_B (x * (\sigma(r) * \sigma(s) + \sigma(s) + \sigma(s)) + \sigma(s) + \sigma(s)) \\
&\approx_B (x * (\sigma(r * s)) + 0) \\
&\approx_B 0
\end{aligned}$$

Now we show that σ' is also reproductive. Let τ be a B-unifier of $t \stackrel{?}{\approx}_B 0$. Because σ is a reproductive B-unifier of $t' \stackrel{?}{\approx}_B 0$ and (1) implies that τ is indeed a B-unifier of $t' \stackrel{?}{\approx}_B 0$ it follows that $\tau(\sigma(y)) \approx_B \tau(y)$ for all y. Therefore $\tau(\sigma'(y)) = \tau(\sigma(y)) \approx_B \tau(y)$ for all $y \neq x$. For $y = x$:

$$\begin{aligned}
\tau(\sigma'(x)) = \quad & \tau(x * (\sigma(r) + \sigma(s) + 1) + \sigma(s)) \\
\approx_B \ & \tau(x) * (\tau(\sigma(r)) + \tau(\sigma(s)) + 1) + \tau(\sigma(s)) \\
\approx_B \ & \tau(x) * (\tau(r) + \tau(s) + 1) + \tau(s) \\
\approx_B \ & \tau(x) * \tau(r) + \tau(x) * \tau(s) + \tau(x) + \tau(s) \\
\approx_B \ & \tau(x) * \tau(r) + (\tau(x) + 1) * \tau(s) + \tau(x) \\
\approx_B \ & \tau(t) + \tau(x) \\
\approx_B \ & \tau(x)
\end{aligned}$$

□

Now lets consider as example the B-unification problem $xy+yz+xz+1 \stackrel{?}{\approx}_B 0$. First we eliminate x.

$$x*(y+z+yz+1)+(x+1)*(yz+1) \stackrel{?}{\approx}_B 0 \ (r=y+z+yz+1, s=yz+1)$$
$$r*s \approx_B yz+yz+yz+yz+y+z+yz+1 \approx_B y+z+yz+1$$

Now we eliminate y.

$$y*(1+z+z+1)+(y+1)*(z+1) \stackrel{?}{\approx}_B 0 \ (r'=1+z+z+1, s'=z+1)$$
$$r'*s' \approx_B (1+z+z+1)*(z+1) \approx_B 0*(z+1) \approx_B 0$$

Obviously $0 \stackrel{?}{\approx} 0$ is B-unifiable with the reproductive B-unifier σ'' which is the identity on all variables. From theorem 3.5 it follows that σ' defined by

$$\sigma' := \sigma'' \cup \{y \mapsto y*(\sigma''(r')+\sigma''(s')+1)+\sigma''(s')\}$$
$$\sigma' := \sigma'' \cup \{y \mapsto y*(0+z+1+1)+z+1\}$$
$$\sigma' := \sigma'' \cup \{y \mapsto yz+z+1\}$$

is a reproductive B-unifier of $y*(1+z+z+1)+(y+1)*(z+1) \stackrel{?}{\approx}_B 0$. Correspondingly σ defined by

$$\sigma := \sigma' \cup \{x \mapsto x*(\sigma'(r+s)+1)+\sigma'(s)\}$$
$$\sigma := \sigma' \cup \{x \mapsto x*(\sigma'(y+z)+1)+(yz+z+1)*z+1\}$$
$$\sigma := \sigma' \cup \{x \mapsto x*(yz+z+1+z+1)+yz+1\}$$
$$\sigma := \sigma' \cup \{x \mapsto xyz+yz+1\}$$

is a reproductive B-unifier of $x*(y+z+yz+1)+(x+1)*(yz+1) \stackrel{?}{\approx}_B 0$ and also of our original problem $xy+yz+xz+1 \stackrel{?}{\approx}_B 0$.

3.4 Löwenheim's formula

This algorithm is not as intuitive as successive variable elimination but much more interesting. The idea is to turn an arbitrary B-unifier τ of $t \stackrel{?}{\approx}_B 0$ into an mgu (even a reproductive B-unifier).

Theorem 3.6. *Let τ be a B-unifier of $t \stackrel{?}{\approx}_B 0$. The substitution σ defined by*

$$\sigma(x) := \begin{cases} (t+1)*x+t*\tau(x) & \text{if } x \in \mathcal{V}ar(t) \\ x & \text{if } x \notin \mathcal{V}ar(t) \end{cases}$$

is a reproductive B-unifier of $t \stackrel{?}{\approx}_B 0$.

Before we can proof this theorem we need the following lemma to proof that σ actually is a B-unifier.

Lemma 3.7. *If* $\sigma(x) = (s+1) * \sigma_1(x) + s * \sigma_2(x)$ *for all* $x \in \mathcal{V}ar(t)$*, then* $\sigma(t) = (s+1) * \sigma_1(t) + s * \sigma_2(t)$.

Proof. We show this by structural induction on t.

$t = x$: The base case is trivial:

$$\begin{aligned}\sigma(t) &= (s+1) * \sigma_1(x) + s * \sigma_2(x) \\ &= (s+1) * \sigma_1(t) + s * \sigma_2(t)\end{aligned}$$

$t = 0$:

$$\begin{aligned}\sigma(t) = 0 &\approx_B 0 + 0 \approx_B (s+1) * 0 + s * 0 \\ &= (s+1) * \sigma_1(t) + s * \sigma_2(t)\end{aligned}$$

$t = 1$:

$$\begin{aligned}\sigma(t) = 1 &\approx_B s + 1 + s \approx_B (s+1) * 1 + s * 1 \\ &= (s+1) * \sigma_1(t) + s * \sigma_2(t)\end{aligned}$$

$t = t_1 + t_2$: Using the induction hypothesis we obtain:

$$\begin{aligned}\sigma(t) = &\quad \sigma(t_1) + \sigma(t_2) \\ &\approx_B (s+1) * \sigma_1(t_1) + s * \sigma_2(t_1) + (s+1) * \sigma_1(t_2) + s * \sigma_2(t_2) \\ &\approx_B (s+1) * (\sigma_1(t_1) + \sigma_1(t_2)) + s * (\sigma_2(t_1) + \sigma_2(t_2)) \\ &= \quad (s+1) * \sigma_1(t) + s * \sigma_2(t)\end{aligned}$$

$t = t_1 * t_2$: Using the induction hypothesis we obtain:

$$\begin{aligned}\sigma(t) = &\quad \sigma(t_1) * \sigma(t_2) \\ &\approx_B ((s+1) * \sigma_1(t_1) + s * \sigma_2(t_1)) * ((s+1) * \sigma_1(t_2) + s * \sigma_2(t_2)) \\ &\approx_B (s+1) * \sigma_1(t_1) * \sigma_1(t_2) + 0 + 0 + s * \sigma_2(t_1) * \sigma_2(t_2) \\ &= \quad (s+1) * \sigma_1(t) + s * \sigma_2(t)\end{aligned}$$

□

Now we can proof theorem 3.6 easily.

Proof(Theorem 3.6). At first we need to show that σ is indeed a B-unifier of $t \overset{?}{\approx}_B 0$. This is easy since σ has exactly the form we need for lemma 3.7 ($s := t, \sigma_1(x) := x$ for all x and $\sigma_2 := \tau$). The fact that τ is a B-unifier of $t \overset{?}{\approx}_B 0$ now gives us:

$$\sigma(t) = (t+1) * t + t * \tau(t) \approx_B 0 + 0 \approx_B 0$$

So σ is a B-unifier, it is easy to show that it is also reproductive. Let τ' be an arbitrary B-unifier of $t \overset{?}{\approx}_B 0$. If $x \in \mathcal{V}ar(t)$ then

$$\begin{aligned}\tau'(\sigma(x)) = &\quad \tau'((t+1) * x + t * \tau(x)) \\ = &\quad (\tau'(t)+1) * \tau'(x) + \tau'(t) * \tau'(\tau(x)) \\ \approx_B &\ (0+1) * \tau'(x) + 0 * \tau'(\tau(x)) \\ \approx_B &\ \tau'(x)\end{aligned}$$

if $x \notin \mathcal{V}ar(t)$ then $\sigma(x) = x$ so $\tau'(\sigma(x)) = \tau'(x)$ follows trivially. □

Now lets consider the B-unification problem $xy + yz + xz + 1 \overset{?}{\approx}_B 0$ (the same as in successive variable elimination) as example. We will only consider ground B-unifiers.

$$\begin{aligned}xyz = 1: \qquad \sigma_1(x) = &\quad (xy+yz+xz+1+1) * x + (xy+yz+xz+1) * 1 \\ \approx_B &\ xy + xyz + xz + (xy + yz + xz + 1) \\ \approx_B &\ xyz + yz + 1 \\ \sigma_1(y) = &\quad (xy+yz+xz+1+1) * y + (xy+yz+xz+1) * 1 \\ \approx_B &\ xyz + xz + 1 \\ \sigma_1(z) = &\quad (xy+yz+xz+1+1) * z + (xy+yz+xz+1) * 1 \\ \approx_B &\ xyz + xy + 1 \\ x = 0, yz = 1: \quad \sigma_2(x) = &\quad (xy+yz+xz+1+1) * x + (xy+yz+xz+1) * 0 \\ \approx_B &\ xy + xyz + xz \\ \sigma_2(y) \approx_B &\ xyz + xz + 1 \\ \sigma_2(z) \approx_B &\ xyz + xy + 1\end{aligned}$$

$x = 1, y = 0, z = 1$ and $x = 1, y = 1, z = 0$ are symettric to $x = 0, y = 1, z = 1$

Since it is not so easy to see that the calculated B-unifiers are reproductive we will consider an example. Remember the reproductive B-unifier $\sigma := \{x \mapsto xyz + yz + 1, y \mapsto yz + z + 1\}$ which we computed with successive variable elimination. Lets show that $\sigma(\sigma_1(w)) \approx_B \sigma(w)$ for all w. If $w \notin \{x, y, z\}$

then $\sigma(\sigma_1(w)) = \sigma(w)$ follows trivially. If $w \in \{x, y, z\}$ then

$$\begin{aligned}
\sigma(\sigma_1(x)) &\approx_B \sigma(xyz + yz + 1) \\
&\approx_B (xyz + yz + 1)(yz + z + 1)z + \sigma(yz + 1) \\
&\approx_B (xyz + yz + yz) + (xyz + yz + z) + (xyz + yz + z) + \sigma(yz + 1) \\
&\approx_B xyz + (yz + z + 1)z + 1 \\
&\approx_B xyz + yz + 1 \\
&\approx_B \sigma(x) \\
\sigma(\sigma_1(y)) &\approx_B \sigma(xyz + xz + 1) \\
&\approx_B xyz + \sigma(xz + 1) \\
&\approx_B xyz + (xyz + yz + 1)z + 1 \\
&\approx_B yz + z + 1 \\
&\approx_B \sigma(y)
\end{aligned}$$

$$\begin{aligned}
\sigma(\sigma_1(z)) &\approx_B \sigma(xyz + xy + 1) \\
&\approx_B xyz + \sigma(xy + 1) \\
&\approx_B xyz + (xyz + yz + 1)(yz + z + 1) + 1 \\
&\approx_B xyz + (xyz + yz + yz) + (xyz + yz + z) + (xyz + yz + 1) + 1 \\
&\approx_B z \\
&\approx_B \sigma(z)
\end{aligned}$$

Of course this does not proof that σ_1 is reproductive it was just to stress the fact that for a reproductive B-unifier σ the equation $\tau(\sigma(x)) \approx_B \tau(x)$ has to hold for all B-unifiers τ so it especially has to hold for other reproductive B-unifiers.

3.5 Conclusion

Since for every B-unification problem we could just put the corresponding boolean formula into an SAT-solver to get an B-unifier for Löwenheims's formula it is easy to see that computing an mgu with Löwenheims's formula takes nondeterministic polynomial time. For successive variable elimination this is not so easy to show. But since by finding an mgu we also solve the SAT problem for boolean formulas it has to be NP-hard. Keep in mind that we have only considered elementary B-unification problems here, B-unification problems with constants and general B-unification problems have higher complexity [2].

References

[1] Franz Baader and Tobias Nipkow, *Term Rewriting and All That.* Cambridge University Press, 1st edition, 1999.

[2] Franz Baader, *On the complexity of Boolean unification.* Information Processing Letters 67, pages 215-220, 1998.