Free Variables and Theories: Revisiting Rigid E-Unification[⋆]

Peter Backeman and Philipp Rümmer

Uppsala University, Sweden

Abstract. The efficient integration of theory reasoning in first-order calculi with free variables (such as sequent calculi or tableaux) is a long-standing challenge. For the case of the theory of equality, an approach that has been extensively studied in the 90s is rigid E-unification, a variant of equational unification in which the assumption is made that every variable denotes exactly one term (rigid semantics). The fact that simultaneous rigid E-unification is undecidable, however, has hampered practical adoption of the method, and today there are few theorem provers that use rigid E-unification.
One solution is to consider incomplete algorithms for computing (simultaneous) rigid E-unifiers, which can still be sufficient to create sound and complete theorem provers for first-order logic with equality; such algorithms include rigid basic superposition proposed by Degtyarev and Voronkov, but also the much older subterm instantiation approach introduced by Kanger in 1963 (later also termed minus-normalisation). We introduce bounded rigid E-unification (BREU) as a new variant of E-unification corresponding to subterm instantiation. In contrast to general rigid E-unification, BREU is NP-complete for individual and simultaneous unification problems, and can be solved efficiently with the help of SAT; BREU can be combined with techniques like congruence closure for ground reasoning, and be used to construct theorem provers that are competitive with state-of-the-art tableau systems. We outline ongoing research how BREU can be generalised to other theories than equality.

1 Introduction

The integration of efficient equality reasoning, and theory reasoning in general, in tableaux and sequent calculi is a long-standing challenge, and has led to a wealth of theoretically intriguing, yet surprisingly few practically satisfying solutions. Among others, a family of approaches related to the (undecidable) problem of computing *simultaneous rigid* E*-unifiers* have been developed, by utilising incomplete unification procedures in such a way that an overall complete first-order calculus is obtained [11, 4, 9]. Following the line of research started by Kanger [11], we recently introduced *simultaneous bounded rigid* E*-unification* (BREU) [2], a new version of rigid E-unification that is bounded in the sense that

[⋆] This work was partly supported by the Microsoft PhD Scholarship Programme and the Swedish Research Council.

variables only represent terms from finite domains, thus preserving decidability even for simultaneous E-unification problems. As demonstrated in [2], BREU can be used to design sound and complete calculi for first-order logic with equality, and to implement theorem provers that compare favourably to state-of-the-art tableau provers in terms of performance on problems with equality.

In this paper, we study the problem of generalising from BREU to bounded rigid unification modulo theories beyond equality. To this end, we first investigate different ways of defining semantics of BREU problems: BREU problems can be interpreted both syntactically and semantically, leading to two formalisms that differ in terms of expressiveness and complexity. We discuss how the semantic setting lends itself to generalisation rather naturally, in particular for theories that admit quantifier elimination. We conclude by outlining resulting challenges.

2 Background

2.1 Rigid *E*-Unification

We start by illustrating the rigid E-unification approach using the following problem from [4]:

$$\phi \quad = \quad \exists x, y, u, v. \begin{pmatrix} (a \approx b \;\rightarrow\; g(x,u,v) \approx g(y, f(c), f(d))) \;\wedge \\ (c \approx d \;\rightarrow\; g(u,x,y) \approx g(v, f(a), f(b))) \end{pmatrix}$$

To show validity of ϕ, a Gentzen-style proof (or, equivalently, a tableau) can be constructed, using free variables for x, y, u, v:

$$\dfrac{\dfrac{\dfrac{\mathcal{A}}{a \approx b \;\vdash\; g(X,U,V) \approx g(Y, f(c), f(d))} \qquad \dfrac{\mathcal{B}}{c \approx d \;\vdash\; g(U,X,Y) \approx g(V, f(a), f(b))}}{\vdash\; (a \approx b \rightarrow g(X,U,V) \approx g(Y, f(c), f(d))) \wedge (c \approx d \rightarrow g(U,X,Y) \approx g(V, f(a), f(b)))}}{\vdash\; \phi}$$

To finish this proof, both $\mathcal{A}$ and $\mathcal{B}$ need to be closed by applying further rules, and substituting concrete terms for the variables. The substitution $\sigma_l = \{X \mapsto Y, U \mapsto f(c), V \mapsto f(d)\}$ makes it possible to close $\mathcal{A}$ through equational reasoning, and $\sigma_r = \{X \mapsto f(a), U \mapsto V, Y \mapsto f(b)\}$ closes $\mathcal{B}$, but neither closes both. Finding a substitution that closes both branches is known as *simultaneous rigid* E*-unification* (SREU), and has first been formulated in [8]:

Definition 1 (Rigid *E*-Unification). *Let E be a set of equations, and s, t be terms. A substitution σ is called a* rigid E-unifier *of s and t if $s\sigma \approx t\sigma$ follows from $E\sigma$ via ground equational reasoning. A* simultaneous rigid E-unifier *σ is a common rigid* E*-unifier for a set $(E_i, s_i, t_i)_{i=1}^n$ of rigid* E*-unification problems.*

In our example, two rigid E-unification problems have to be solved:

$$\begin{aligned} E_1 &= \{a \approx b\}, \qquad & s_1 &= g(X,U,V), \quad & t_1 &= g(Y, f(c), f(d)), \\ E_2 &= \{c \approx d\}, \qquad & s_2 &= g(U,X,Y), \quad & t_2 &= g(V, f(a), f(b)). \end{aligned}$$

We can observe that $\sigma_s = \{X \mapsto f(a), Y \mapsto f(b), U \mapsto f(c), V \mapsto f(d)\}$ is a simultaneous rigid E-unifier, and suffices to finish the proof of ϕ.

The SREU problem famously turned out undecidable [3], which makes the style of reasoning shown here problematic in automated theorem provers. Different solutions have been proposed to address this situation, including potentially non-terminating, but complete E-unification procedures [7], and terminating but incomplete algorithms that are nevertheless sufficient to create complete proof procedures [11, 4, 9]. The practical impact of such approaches has been limited; to the best of our knowledge, there is no (at least no actively maintained) theorem prover based on such explicit forms of SREU.

2.2 Subterm Instantiation and Bounded Rigid E-Unification

An early solution in the class of "terminating, but incomplete" algorithms for SREU was introduced as *dummy instantiation* in the seminal work of Kanger [11] (in 1963, i.e., even before the introduction of unification), and later studied under the names *subterm instantiation* and *minus-normalisation* [5, 6]; the relationship to SREU was observed in [4]. In contrast to full SREU, subterm instantiation only considers E-unifiers where substituted terms are taken from some predefined finite set, which directly implies decidability. The impact of subterm instantiation on practical theorem proving was again limited, however, among others because no efficient search procedures for dummy instantiation were available [6].

In recent work, we have introduced *bounded rigid* E-*unification* (BREU), a new restricted version of SREU that captures the decision problem to be solved in the subterm instantiation method, and developed symbolic algorithms for computing bounded rigid E-unifiers [2, 1]. We illustrate the application of BREU on the example from the previous section; for sake of presentation, BREU operates on formulae that are normalised by means of flattening (observe that ϕ and ϕ' are equivalent):

$$\begin{aligned}\phi' \;=\; & \forall z_1, z_2, z_3, z_4.\ \big(f(a) \not\approx z_1 \lor f(b) \not\approx z_2 \lor f(c) \not\approx z_3 \lor f(d) \not\approx z_4 \lor \\ & \qquad \exists x, y, u, v.\ \forall z_5, z_6, z_7, z_8.\ \begin{pmatrix} g(x,u,v) \not\approx z_5 \lor g(y, z_3, z_4) \not\approx z_6 \lor \\ g(u,x,y) \not\approx z_7 \lor g(v, z_1, z_2) \not\approx z_8 \lor \\ ((a \not\approx b \lor z_5 \approx z_6) \land (c \not\approx d \lor z_7 \approx z_8)) \end{pmatrix}\big)\end{aligned}$$

A proof constructed for ϕ' has the same structure as the one for ϕ, with the difference that all function terms are now isolated in the antecedent:

$$\dfrac{\dfrac{\dfrac{\mathcal{A}'}{\ldots, g(X,U,V) \approx o_5, a \approx b \ \vdash\ o_5 \approx o_6} \qquad \dfrac{\mathcal{B}'}{\ldots, g(U,X,Y) \approx o_7, c \approx d \ \vdash\ o_7 \approx o_8}}{\vdots}}{\dfrac{\overline{f(a) \approx o_1 \lor f(b) \approx o_2 \lor f(c) \approx o_3 \lor f(d) \approx o_4 \ \vdash\ \exists x, y, u, v.\ \forall z_5, z_6, z_7, z_8.\ \ldots}\ (*)}{\dfrac{\vdots}{\vdash\ \forall z_1, z_2, z_3, z_4.\ \ldots}}}$$

To obtain a *bounded* rigid E-unification problem, we now restrict the terms considered for instantiation of X, Y, U, V to the symbols that were in scope when the variables were introduced (at $(*)$ in the proof): X ranges over constants $\{o_1, o_2, o_3, o_4\}$, Y over $\{o_1, o_2, o_3, o_4, X\}$, and so on. Since the problem is flat, those sets contain representatives of all existing ground terms at point $(*)$ in the proof. It is therefore possible to find a simultaneous E-unifier, namely the substitution $\sigma_b = \{X \mapsto o_1, Y \mapsto o_2, U \mapsto o_3, V \mapsto o_4\}$.

Despite the restriction to terms of only bounded size, the subterm instantiation strategy gives rise to a sound and complete calculus for first-order logic with equality [2]; intuitively, the calculus will eventually generate all required terms by repeated instantiation of quantified formulae. The finiteness of considered BREU problems, at any point during proof search, enables the use of efficient techniques from the SAT and SMT domain to check for the existence of unifiers.

2.3 Bounded Rigid E-Unification Formally

Given countably infinite sets C of constants (denoted by $c, d, \dots$), V_b of bound variables (written $x, y, \dots$), and V of free variables (denoted by $X, Y, \dots$), as well as a finite set F of fixed-arity function symbols (written $f, g, \ldots$), the syntactic categories of *formulae* ϕ and *terms* t are defined by

$$\phi \ ::= \ \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \forall x.\phi \mid \exists x.\phi \mid t \approx t \ , \qquad t \ ::= \ c \mid x \mid X \mid f(t, \dots, t) \ .$$

We sometimes write $\phi \to \psi$ as shorthand notation for $\neg\phi \vee \psi$, and generally assume that bound variables x only occur underneath quantifiers $\forall x$ or $\exists x$. Semantics of terms and formulae without free variables is defined as is common using first-order structures (U, I) consisting of a non-empty universe U, and an interpretation function I.

We call constants and (free or bound) variables *atomic terms*, and all other terms *compound terms*. A *flat equation* is an equation between atomic terms, or an equation of the form $f(t_1, \dots, t_n) \approx t_0$, where $t_0, \dots, t_n$ are atomic terms. A substitution is a mapping of variables to terms, such that all but finitely many variables are mapped to themselves. Symbols $\sigma, \theta, \dots$ denote substitutions, and we use post-fix notation $\phi\sigma$ or $t\sigma$ to denote application of substitutions. An *atomic substitution* is a substitution that maps variables only to atomic terms. We write $u[r]$ do denote that r is a subexpression of a term or formula u, and $u[s]$ for the term or formula obtained by replacing the subexpression r with s.

Definition 2 (Replacement relation [13]). *The* replacement relation $\to_E$ *induced by a set of equations E is defined by: $u[l] \to u[r]$ if $l \approx r \in E$. The relation $\leftrightarrow_E^*$ represents the reflexive, symmetric and transitive closure of $\to_E$.*

Definition 3 (BREU). *A* bounded rigid E-unification (BREU) problem *is a triple $(\preceq, E, e)$, with $\preceq$ being a partial order over atomic terms such that for all variables X the set $\{s \mid s \preceq X\}$ is finite; E is a finite set of flat formulae; and $e = s \approx t$ is an equation between atomic terms (the target equation). An atomic substitution σ is called a* bounded rigid E-unifier *of s and t if $s\sigma \leftrightarrow_{E\sigma}^* t\sigma$ and $X\sigma \preceq X$ for all variables X.*

Definition 4 (Simultaneous BREU). *A* simultaneous bounded rigid E-unification problem *is a pair* $(\preceq, (E_i, e_i)_{i=1}^n)$ *such that each triple* $(\preceq, E_i, e_i)$ *is a bounded rigid* E*-unification problem. A substitution* σ *is a* simultaneous bounded rigid E-unifier *if it is a bounded rigid* E*-unifier for each problem* $(\preceq, E_i, e_i)$*.*

In the following, we say that a (possibly simultaneous) BREU problem is *syntactically solvable* if a bounded rigid E-unifier exists. As has been shown in [2], checking syntactic solvability is NP-hard, and can effectively be solved via an encoding to SAT, or with SMT-style reasoning.

Example 5. We revisit the example introduced in Sect. 2.1, which can be captured as the following simultaneous BREU problem $(\preceq, \{(E_1, e_1), (E_2, e_2)\})$:

$$E_1 = E \cup \{a \approx b\}, \quad e_1 = o_5 \approx o_6, \qquad E_2 = E \cup \{c \approx d\}, \quad e_2 = o_7 \approx o_8,$$

$$E = \left\{ \begin{array}{l} f(a) \approx o_1, f(b) \approx o_2, f(c) \approx o_3, f(d) \approx o_4, \\ g(X, U, V) \approx o_5, g(Y, o_3, o_4) \approx o_6, g(U, X, Y) \approx o_7, g(V, o_1, o_2) \approx o_8 \end{array} \right\}$$

with $a \prec b \prec c \prec d \prec o_1 \prec o_2 \prec o_3 \prec o_4 \prec X \prec Y \prec U \prec V \prec o_5 \prec o_6 \prec o_7 \prec o_8$.

A unifier for this problem is sufficient to close all goals of the tree up to equational reasoning; one solution is $\sigma = \{X \mapsto o_1, Y \mapsto o_2, U \mapsto o_3, V \mapsto o_4\}$.

The remainder of the paper considers the question how the notion of BREU can be carried over to other theories than just equality. As we will see, to this end it is useful to provide a more relaxed characterisation of BREU solvability.

3 Semantically Solving BREU

Definition 6 (Forest-shaped BREU). *A BREU problem* $(\preceq, (E_i, e_i)_{i=1}^n)$ *is* forest-shaped *if (i) the order* $\preceq$ *forms a forest, that is, whenever* $a \preceq b$ *and* $a' \preceq b$ *it is the case that* $a \preceq a'$ *or* $a' \preceq a$*; and (ii) components* (E_i, e_i) *(for* $i \in \{1, \ldots, n\}$*) do not mix atomic terms from several branches of* $\preceq$*, that is, whenever* (E_i, e_i) *contains atomic terms* s, t *it is the case that* $s \preceq t$ *or* $t \preceq s$*.*

The BREU problem given in Example 5, and generally all BREU problems extracted from proofs (as defined in [2]) are forest-shaped; the structure of $\preceq$ will reflect the proof tree from which the BREU problem was derived. Importantly, forest-shaped problems can be reinterpreted as formulae by translating the order $\preceq$ to a prefix of quantifiers, and replacing equations E_i with Ackermann constraints. Without loss of generality, we assume that every equation in a set E_i of a BREU problem $(\preceq, (E_i, e_i)_{i=1}^n)$ contains a function symbol; equations $a \approx b$ between constants or variables can be rewritten to $f() \approx a, f() \approx b$ by introducing a fresh zero-ary function f.

Definition 7 (BREU formula encoding). *Suppose* $B = (\preceq, (E_i, s_i \approx t_i)_{i=1}^n)$ *is a forest-shaped simultaneous BREU problem, and* S *the (finite) set of atomic terms occurring in* B*, with* $k = |S|$*. Let further* $S_b = \{x_1, \ldots, x_k\} \subseteq V_b$ *be fresh bound variables (not occurring in* B*), and* $\sigma : S \to S_b$ *a bijection such that* $\sigma(s) =$

x_i, $\sigma(t) = x_j$ and $s \preceq t$ imply $i \leq j$. Then the formula $Q_1 x_1. \dots .Q_k x_k.\ \bigwedge_{j=1}^n G_j$ with

$$Q_i = \begin{cases} \forall & \text{if } \sigma^{-1}(x_i) \in C \text{ is a constant} \\ \exists & \text{otherwise} \end{cases}$$

$$G_j = \Big(\bigwedge_{f(\bar{a}) \approx b, f(\bar{a}') \approx b' \in E_j} \sigma(\bar{a}) \approx \sigma(\bar{a}') \to \sigma(b) \approx \sigma(b') \Big) \to \sigma(s_j) \approx \sigma(t_j)$$

is called a formula encoding *of B.*

Example 8. Consider the BREU problem $B = (\preceq, E, e)$ defined by

$$E = \{f(X) \approx c, f(a) \approx a, f(b) \approx b\}, \qquad e = a \approx b, \qquad a \prec b \prec c \prec X\ .$$

To encode B as a formula, we fix fresh variables $x_1, \dots, x_4$ and the mapping $\sigma = \{a \mapsto x_1, b \mapsto x_2, c \mapsto x_3, X \mapsto x_4\}$, and obtain

$$\forall x_1. \forall x_2. \forall x_3. \exists x_4.\ (x_4 \approx x_1 \to x_3 \approx x_1) \land (x_4 \approx x_2 \to x_3 \approx x_2) \to x_1 \approx x_2\ . \quad (1)$$

Here, the assumption $x_4 \approx x_1 \to x_3 \approx x_1$ stems from the Ackermann constraint $X \approx a \to c \approx a$, and $x_4 \approx x_2 \to x_3 \approx x_2$ from $X \approx b \to c \approx b$; other Ackermann constraints are either tautologies, or equivalent to the two constraints given, and have been left out for sake of brevity.

The formula encoding of a BREU problem is a first-order formula with equality, but without functions symbols; the validity of the encoding is therefore decidable. It can also be observed that Def. 7 in principle admits multiple formula encodings for a BREU problem, but those different encodings are guaranteed to be equivalent, thanks to the fact that the BREU problem is forest-shaped.

We say that a BREU problem is *semantically solvable* if its formula encoding is valid. Semantic solvability is a weaker property than syntactic solvability (as in Def. 3). In particular, it can easily be checked that (1) is valid, but the problem B from Example 8 does not have any syntactic E-unifiers: such a unifier would have to map X to one of X, a, b, c, but in no case it is possible to conclude $a \approx b$.

Lemma 9. *If a (possibly simultaneous) forest-shaped BREU problem B has an* E*-unifier, then the formula encoding of B is valid.*

Proof. Any syntactic E-unifier defines how existential quantifiers in the formula encoding have to be instantiated to satisfy the formula. □

3.1 Semantic Solvability in a First-order Calculus

The sequent calculus for first-order logic with equality introduced in [2] uses BREU to implement a global closure rule for free-variable proofs:

$$\dfrac{\dfrac{*}{\Gamma_1 \vdash \Delta_1} \quad \cdots \quad \dfrac{*}{\Gamma_n \vdash \Delta_n}\ \text{BREU}}{\Gamma \vdash \Delta}$$

where $\Gamma_1 \vdash \Delta_1, \dots, \Gamma_n \vdash \Delta_n$ are all open goals of the proof, $E_i = \{t \approx s \in \Gamma_i\}$ are flat antecedent equations, $e_i = \bigvee\{t \approx s \in \Delta_i\}$ are succedent equations, and the simultaneous BREU problem $(\preceq, (E_i, e_i)_{i=1}^n)$ is solvable

The rule uses a slightly generalised version of BREU in which a target constraint e_i can be a disjunction of equations; such problems can easily be translated to normal BREU at the cost of introducing additional function symbols. The order $\preceq$ in the rule is derived from the structure of a proof, and the BREU problem $(\preceq, (E_i, e_i)_{i=1}^n)$ is in particular forest-shaped. Given the alternative notion of semantic solvability, the question arises whether overall soundness and completeness of the first-order calculus from [2] are preserved when reformulating the BREU rule to be applicable whenever "the simultaneous BREU problem $(\preceq, (E_i, e_i)_{i=1}^n)$ is *semantically* solvable." We will call the new rule BREU$_{\text{SEM}}$.

The answer is positive in both cases. From Lem. 9, it follows directly that replacing BREU with BREU$_{\text{SEM}}$ preserves completeness of the calculus, because the weaker side condition only entails that BREU$_{\text{SEM}}$ might be applicable in more cases than BREU. Soundness cannot be concluded from the soundness proof given in [2] for the syntactic case, but we can instead find a simple inductive argument that the formula encoding of the BREU problem $(\preceq, (E_i, e_i)_{i=1}^n)$ constructed in BREU$_{\text{SEM}}$ is always an *under-approximation* of the root sequent of a proof. Thus, if the formula encoding is valid, also the validity of the root sequent follows:

Lemma 10. *Suppose $\Gamma \vdash \Delta$ is a sequent without free variables, and P a proof constructed from $\Gamma \vdash \Delta$. If B is the BREU problem constructed in an application of* BREU$_{\text{SEM}}$ *to P, then the formula encoding ϕ_B of B implies $\bigwedge \Gamma \to \bigvee \Delta$.*

3.2 The Complexity of Semantic Solvability

Example 8 illustrates that the notion of semantic solvability does not coincide with (and is therefore strictly weaker than) syntactic solvability; this implies that the use of the relaxed rule BREU$_{\text{SEM}}$ can sometimes lead to *shorter proofs* compared to the original rule BREU. The resulting gain in efficiency is offset, however, by the increased computational complexity of checking BREU solvability: in the syntactic case, this problem is NP-complete [2], whereas it turns out that semantic solvability is *PSPACE-complete.* For membership in PSPACE, observe that the formula encoding of a BREU problem can directly be mapped to a Quantified Boolean Formula (QBF), since it is only necessary to consider universes with as many individuals as the formula contains quantifiers.

Lemma 11 (PSPACE-hardness). *Checking whether a (possibly simultaneous) forest-shaped BREU problem has a valid formula encoding is PSPACE-hard.*

Proof. We show that QBF formulae ϕ can be translated to BREU problems B_ϕ, in such a way that the formula encoding of B_ϕ is equivalent to ϕ. Assume that $\phi = Q_1 x_1. \ldots .Q_k x_k.\psi$ is in prenex normal form, with $Q_i \in \{\exists, \forall\}$, and ψ is a Boolean formula over the variables $x_1, \ldots, x_k$ and connectives $\neg, \vee$.

To represent truth values, two constants $\mathbf{0}, \mathbf{1} \in C$ are introduced. Then, to handle the quantifiers, for each variable x_i with $Q_i = \exists$ a fresh free variable $X_i \in V$ is picked, and for each x_i with $Q_i = \forall$ a fresh variable $X_i \in V$

and a fresh constant $d_i \in C$. In addition, in the latter case we define two BREU sub-problems (E_i^0, e_i^0) and (E_i^1, e_i^1) with

$$E_i^0 = \{d_i \approx \mathbf{0}\}, \qquad e_i^0 = X_i \approx \mathbf{0}, \qquad E_i^1 = \{d_i \approx \mathbf{1}\}, \qquad e_i^1 = X_i \approx \mathbf{1} \ .$$

To represent the Boolean structure of ψ, like in [2] two function symbols f_{or} and f_{not} are introduced, which are axiomatised by equations $E_{\mathbb{B}} = \{f_{or}(\mathbf{0},\mathbf{0}) \approx \mathbf{0}, f_{or}(\mathbf{0},\mathbf{1}) \approx \mathbf{1}, f_{or}(\mathbf{1},\mathbf{0}) \approx \mathbf{1}, f_{or}(\mathbf{1},\mathbf{1}) \approx \mathbf{1}, f_{not}(\mathbf{0}) \approx \mathbf{1}, f_{not}(\mathbf{1}) \approx \mathbf{0}\}$. Each sub-formula θ of ψ is then encoded using a fresh constant c_θ and an equation e_θ:

$$\begin{aligned}
e_\theta &= X_i \approx c_\theta && \text{if } \theta = x_i,\\
e_\theta &= f_{not}(c_{\theta_1}) \approx c_\theta && \text{if } \theta = \neg\theta_1,\\
e_\theta &= f_{or}(c_{\theta_1}, c_{\theta_2}) \approx c_\theta && \text{if } \theta = \theta_1 \vee \theta_2 \ .
\end{aligned}$$

We write $E_\psi = \{e_\theta \mid \theta \text{ a sub-formula of } \psi\}$ for the set of all such equations.

Finally, the resulting (forest-shaped) BREU problem is

$$B_\phi = (\preceq, \{(E_i^0, e_i^0), (E_i^1, e_i^1) \mid i \in \{1, \ldots, k\}, Q_i = \forall\} \cup \{(E_{\mathbb{B}} \cup E_\psi, c_\psi \approx \mathbf{1})\})$$

with a total order $\preceq$ that satisfies $\{\mathbf{0},\mathbf{1}\} \prec \{d_1, X_1\} \prec \cdots \prec \{d_k, X_k\} \prec \{c_\theta \mid \theta \text{ a sub-formula of } \psi\}$ as well as $d_i \prec X_i$ for all $Q_i = \forall$.

To see that ϕ and the formula encoding ϕ_E of B_ϕ are equivalent, we observe that the two formulae have essentially the same quantifier structure, with the difference that (i) ϕ_E starts with quantifiers $\forall x_{\mathbf{0}} \forall x_{\mathbf{1}}$ binding the truth values $\mathbf{0}, \mathbf{1}$; (ii) every quantifier $\exists x_i$ in ϕ is translated to a quantifier $\exists x_{X_i}$ in ϕ_E; (iii) universal quantifiers $\forall x_i$ are translated to $\forall x_{d_i} \exists x_{X_i}$, with the additional goals (E_i^0, e_i^0) and (E_i^1, e_i^1) expressing $x_{d_i} \approx x_{\mathbf{0}} \to x_{X_i} \approx x_{\mathbf{0}}$ and $x_{d_i} \approx x_{\mathbf{1}} \to x_{X_i} \approx x_{\mathbf{1}}$; and (iv) additional $\forall$-quantifiers are added to represent the propositional structure of the matrix ψ. (The somewhat elaborate translation of $\forall x_i$ ensures that universal quantifiers in ϕ_E effectively only range over truth values.) Equivalence of ϕ and ϕ_E follows from the fact that satisfying assignments of the existentially quantified variables can be mapped back and forth between ϕ and ϕ_E. $\square$

4 Towards Bounded Rigid Theory Unification

The notion of semantic solvability offers a natural path to generalise from BREU to Bounded Rigid T-Unification (BRTU), for theories T other than equality. The construction in particular applies to theories that admit quantifier elimination, including various forms of arithmetic. For sake of presentation, we assume that equality $\approx$ is still the only predicate in our logic, but we partition the set $F = F_i \cup F_u$ into a set F_i of interpreted T-functions, and a disjoint set F_u of uninterpreted functions. We further assume that the T-validity of first-order formulae ϕ without uninterpreted functions is decidable.

While the general definition of a BREU problem can be kept as in Def. 3 and 4, we redefine formula encodings to take theory symbols into account:

Definition 12 (BRTU formula encoding). *Suppose $B = (\preceq, (E_i, s_i \approx t_i)_{i=1}^n)$ is a forest-shaped simultaneous BREU problem, and S the (finite) set of atomic terms occurring in B, with $k = |S|$. Let further $S_b = \{x_1, \ldots, x_k\} \subseteq V_b$ be fresh bound variables (not occurring in B), and $\sigma : S \to S_b$ a bijection such that $\sigma(s) = x_i$, $\sigma(t) = x_j$ and $s \preceq t$ imply $i \leq j$. Then the formula $Q_1 x_1. \ldots. Q_k x_k.\ \bigwedge_{j=1}^n G_j^T$ with*

$$Q_i \;=\; \begin{cases} \forall & \textit{if } \sigma^{-1}(x_i) \in C \textit{ is a constant} \\ \exists & \textit{otherwise} \end{cases}$$

$$G_j^T \;=\; \begin{pmatrix} \bigwedge_{\substack{f(\bar{a}) \approx b \in E_j \\ f \in F_i}} f(\sigma(\bar{a})) \approx \sigma(b) \wedge \\ \bigwedge_{\substack{f(\bar{a}) \approx b, f(\bar{a}') \approx b' \in E_j \\ f \in F_u}} \sigma(\bar{a}) \approx \sigma(\bar{a}') \to \sigma(b) \approx \sigma(b') \end{pmatrix} \to \sigma(s_j) \approx \sigma(t_j)$$

*is called a T-*formula encoding *of B.*

Compared to Def. 7, the main change occurs in the definition of G_j^T, where now interpreted functions are kept instead of being replaced with Ackermann constraints. Similarly as before, we say that a BREU problem is *semantically T-solvable* if its T-formula encoding is valid in T.

Example 13. To illustrate the definition, we consider the theory $\mathcal{A}$ of linear (integer or rational) arithmetic, and the implication $f(0) \approx 0 \wedge f(X+1) \approx f(X)+1 \to f(1) \approx 1$, with X ranging over terms $\{0, 1, f(1)\}$. The literals $0, 1$ represent interpreted nullary function symbols, $+$ is an interpreted binary function symbol, and f is an uninterpreted function. Flattening the formula yields a well-formed BREU problem $B = (\preceq, E, e)$ with

$$E = \begin{Bmatrix} 0 \approx c_0, 1 \approx c_1, f(c_0) \approx c_0, f(c_1) = c_2, \\ X + c_1 \approx c_4, f(c_4) \approx c_6, f(X) \approx c_5, c_5 + c_1 \approx c_6 \end{Bmatrix}, \quad e = c_2 \approx c_1,$$
$$c_0 \prec c_1 \prec c_2 \prec X \prec c_4 \prec c_5 \prec c_6 \ .$$

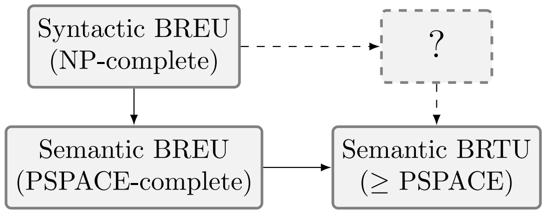
Without taking theory $\mathcal{A}$ into account (treating $0, 1, +$ as uninterpreted functions), B is solvable neither syntactically nor semantically. The $\mathcal{A}$-formula encoding of B is obtained by eliminating f through Ackermann constraints (X is mapped to x_3, and constants c_i to x_i for $i \in \{0, 1, 2, 4, 5, 6\}$; redundant constraints are left out), and is a valid formula in theory $\mathcal{A}$:

$$\begin{array}{l} \forall x_0, x_1, x_2.\ \exists x_3.\ \forall x_4, x_5, x_6. \\ \quad \begin{pmatrix} 0 \approx x_0 \wedge 1 \approx x_1 \wedge x_3 + x_1 \approx x_4 \wedge x_5 + x_1 \approx x_6 \wedge \\ (x_0 \approx x_1 \to x_0 \approx x_2) \wedge (x_0 \approx x_4 \to x_0 \approx x_6) \wedge (x_0 \approx x_3 \to x_0 \approx x_5) \wedge \\ (x_1 \approx x_4 \to x_2 \approx x_6) \wedge (x_1 \approx x_3 \to x_2 \approx x_5) \wedge (x_4 \approx x_3 \to x_6 \approx x_5) \end{pmatrix} \\ \quad \to x_2 \approx x_1 \end{array}$$

5 Challenges and Conclusion

Since Lem. 10 carries over to any of the theories T considered in Sect. 4, the encoding from Def. 12 can in principle be used to implement sound calculi for first-order logic modulo T with bounded free-variable reasoning. For reasons of practicality, of course, various refinements of the overall approach are possible and advisable, along the lines of the procedures presented in [12, 2]; among others, also procedures for ground reasoning in T can be integrated. For the special case of linear integer arithmetic, this style of reasoning was essentially implemented in the theorem prover Princess [12]. There are several more conceptual challenges remaining, however:

Syntactically solving BRTU. We have outlined how solvability of BREU can be characterised semantically, through an encoding as a formula, and then be generalised to theories other than equality. However, both steps have a severe impact on the computational complexity of checking solvability; checking semantic solvability for BRTU modulo linear integer arithmetic, for instance, necessitates a potentially doubly exponential validity check. A crucial question is whether a notion of (theory-dependent) syntactic solvability for BRTU exists, and to investigate the impact on the completeness of an overall proof procedure:



The completeness of proof procedures. It is well-known that no complete calculi exist for first-order logic modulo various theories, for instance modulo linear arithmetic [10]. This leads to the question how the completeness of first-order calculi constructed with the help of BRTU can be characterised, and for which fragments completeness is indeed achieved. The question is partly addressed in [12], but only for linear integer arithmetic and in a setting where uninterpreted functions were replaced with uninterpreted predicates.

References

1. Backeman, P., Rümmer, P.: Efficient algorithms for bounded rigid E-Unification. In: Tableaux. LNCS, Springer (2015), to appear
2. Backeman, P., Rümmer, P.: Theorem proving with bounded rigid E-Unification. In: CADE. LNCS, Springer (2015), to appear
3. Degtyarev, A., Voronkov, A.: Simultaneous rigid E-Unification is undecidable. In: Büning, H.K. (ed.) CSL. LNCS, vol. 1092, pp. 178–190. Springer (1995)
4. Degtyarev, A., Voronkov, A.: What you always wanted to know about rigid E-Unification. J. Autom. Reasoning 20(1), 47–80 (1998)
5. Degtyarev, A., Voronkov, A.: Equality reasoning in sequent-based calculi. In: Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)
6. Degtyarev, A., Voronkov, A.: Kanger's Choices in Automated Reasoning. Springer (2001)

7. Fitting, M.C.: First-Order Logic and Automated Theorem Proving. Graduate Texts in Computer Science, Springer-Verlag, Berlin, 2nd edn. (1996)
8. Gallier, J.H., Raatz, S., Snyder, W.: Theorem proving using rigid e-unification equational matings. In: LICS. pp. 338–346. IEEE Computer Society (1987)
9. Giese, M.: A model generation style completeness proof for constraint tableaux with superposition. In: Tableaux. LNCS, vol. 2381, pp. 130–144. Springer (2002)
10. Halpern, J.Y.: Presburger arithmetic with unary predicates is Π^1_1 complete. Journal of Symbolic Logic 56 (1991)
11. Kanger, S.: A simplified proof method for elementary logic. In: Siekmann, J., Wrightson, G. (eds.) Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966, pp. 364–371. Springer, Berlin, Heidelberg (1983), originally appeared in 1963
12. Rümmer, P.: A constraint sequent calculus for first-order logic with linear integer arithmetic. In: LPAR. LNCS, Springer (2008)
13. Tiwari, A., Bachmair, L., Rueß, H.: Rigid E-Unification revisited. In: CADE. pp. 220–234. CADE-17, Springer-Verlag, London, UK, UK (2000)