Combining Multisets with Integers

Calogero G. Zarba

Stanford University and University of Catania

Abstract. We present a decision procedure for a constraint language combining multisets of ur-elements, the integers, and an arbitrary first-order theory T of the ur-elements. Our decision procedure is an extension of the Nelson-Oppen combination method specifically tailored to the combination domain of multisets, integers, and ur-elements.

1 Introduction

Sets obey the extensionality principle according to which two sets are equal if and only if they contain the same elements, regardless of their multiplicity. In contrast, multisets (also known as bags) are collections of elements in which multiplicity is relevant. Two multisets are equal if and only if they contain the same elements *and* all elements have the same multiplicity in each multiset. Thus, for instance, while the set $\{a, a, b\}$ is considered to be equal to the set $\{a, b\}$, the multiset $[\![a, a, b]\!]$, in which a occurs twice, is different from the multiset $[\![a, b]\!]$, in which a occurs only once.

Multisets appear frequently in mathematics and computer science, and their applications range to combinatorial counting arguments, database query languages, termination proofs, etc.

In this paper we introduce the constraint language **BUI** for expressing constraints over *B*ags, *U*r-elements, and *I*ntegers. Beside a first-order signature Σ_{ur} for expressing constraints over the ur-elements, the language contains:

- the symbols 0, 1, $+$, $-$, max, min and $\leq$ for expressing constraints over integers;
- the empty multiset constant $[\![\,]\!]$, a multiset constructor $[\![\cdot]\!]^{(\cdot)}$, and the multiset operators union ($\cup$), sum ($\uplus$) and intersection ($\cap$) for expressing constraints over multisets;
- a 'count' operator which returns, for each element a and multiset x, the number of occurrences of a in x.

Given an arbitrary theory T modeling the ur-elements, we use a decision procedure that decides the quantifier-free consequences of T and a decision procedure for integer linear arithmetic as black boxes to provide in a modular fashion a decision procedure for the satisfiability of any quantifier-free formula in the language **BUI**.

In addition, we prove that if the decision procedure for the theory T runs in nondeterministic polynomial time, then the satisfiability problem for quantifier-free formulae in the language **BUI** is $\mathcal{NP}$-complete.

Our decision procedure is an extension of the Nelson-Oppen combination method specifically tailored to the combination domain of integers, ur-elements, and multisets. The Nelson-Oppen combination method [1, 3] combines decision procedures for stably infinite[1] first-order theories over disjoint signatures into a single decision procedure for the union theory by means of propagating equalities. It should be noted, however, that our decision procedure for the language **BUI** remains correct even if the underlying theory T of the ur-elements is not stably infinite.

The decision procedure in this paper is inspired by our previous work on the combination of lists with integers [4], and of sets with integers [5].

The paper is organized as follows. In Section 2 we introduce some preliminary notions which will be needed in what follows. In Section 3 we describes our decision procedure, and in Section 4 we prove its correctness. In Section 5 we prove that the satisfiability problem for quantifier-free formulae in the language **BUI** is $\mathcal{NP}$-complete. In Section 6 we discuss some efficiency issues. Finally, in Section 7 we draw conclusions from our work.

2 Preliminaries

In this section we give some basic definitions which will be used in the rest of the paper, and we define the syntax and semantics of the language **BUI.**

2.1 Multisets

Multisets are collections that may contain duplicate elements. Formally, a multiset x is a function $x : A \to \mathbb{N}^+$, for some set A.[2] Given a multiset x and an element a, the number of occurrences of a in x is denoted by $\mathrm{count}(a, x)$, that is:

$$\mathrm{count}(a, x) = \begin{cases} x(a)\,, & \text{if } a \in \mathrm{domain}(x)\,, \\ 0\,, & \text{otherwise.} \end{cases}$$

Equality between two multisets x, y can then be expressed in terms of the count operator:

$$x = y \quad \text{iff} \quad (\forall a)(\mathrm{count}(a, x) = \mathrm{count}(a, y))\,.$$

[1] A first-order theory T is stably infinite if every quantifier-free formula which is satisfiable in T is also satisfiable in an infinite model of T.

[2] Note that according to this definition a multiset $x : A \to \mathbb{N}^+$ can be infinite only if it contains an infinite number of elements, that is, if A is infinite.

Several memberships constructs over multisets can be expressed in terms of the count operator:

$$\begin{aligned}
a \in x &: \ \mathrm{count}(a,x) \geq 1 && (a \text{ is a member of } x)\\
a \notin x &: \ \mathrm{count}(a,x) = 0 && (a \text{ is not a member of } x)\\
a \in^{(n)} x &: \ \mathrm{count}(a,x) \geq n && (a \text{ occurs in } x \text{ at least } n \text{ times})\\
a \in !^{(n)} x &: \ \mathrm{count}(a,x) = n && (a \text{ occurs in } x \text{ exactly } n \text{ times}).
\end{aligned}$$

We use the symbol $[\![\,]\!]$ to denote the empty multiset, whereas we write $[\![a]\!]^{(n)}$ to denote the multiset containing exactly n occurrences of a and nothing else.

Let x, y be two multisets. Then:

- their *union* is the multiset $x \cup y$ such that, for each element a, the equality $\mathrm{count}(a, x \cup y) = \max(\mathrm{count}(a,x), \mathrm{count}(a,y))$ holds;
- their *sum* is the multiset $x \uplus y$ such that, for each element a, the equality $\mathrm{count}(a, x \uplus y) = \mathrm{count}(a,x) + \mathrm{count}(a,y)$ holds;
- their *intersection* is the multiset $x \cap y$ such that, for each element a, the equality $\mathrm{count}(a, x \cap y) = \min(\mathrm{count}(a,x), \mathrm{count}(a,y))$ holds.

2.2 The language BUI: syntax

The language **BUI** is essentially a many-sorted language with three basic sorts bag, ur, and int.

To express constraints over the ur-elements, we are given a first-order signature Σ_{ur}, which we write as $\Sigma^{\mathrm{C}}_{\mathsf{ur}} \cup \Sigma^{\mathrm{F}}_{\mathsf{ur}} \cup \Sigma^{\mathrm{P}}_{\mathsf{ur}}$ where $\Sigma^{\mathrm{C}}_{\mathsf{ur}}$, $\Sigma^{\mathrm{F}}_{\mathsf{ur}}$, $\Sigma^{\mathrm{P}}_{\mathsf{ur}}$ are the collections of constants, function symbols and predicate symbols in Σ_{ur}. In other words, each element of $\Sigma^{\mathrm{C}}_{\mathsf{ur}}$ has sort ur, each element of $\Sigma^{\mathrm{F}}_{\mathsf{ur}}$ of arity n has sort $\underbrace{\mathsf{ur} \times \cdots \times \mathsf{ur}}_{n} \to \mathsf{ur}$, and each element of $\Sigma^{\mathrm{P}}_{\mathsf{ur}}$ of arity n has sort $\underbrace{\mathsf{ur} \times \cdots \times \mathsf{ur}}_{n}$.

To express constraints over the integers, the language **BUI** contains the following symbols:

- the constants 0 (*zero*) and 1 (*one*), both of sort int;
- the operators $+$ (*addition*), $-$ (*subtraction*), max (*maximum*), min (*minimum*), all of them having sort $\mathsf{int} \times \mathsf{int} \to \mathsf{int}$;
- the predicate symbol $\leq$, of sort $\mathsf{int} \times \mathsf{int}$;

To express constraints over multisets, the language **BUI** contains the following symbols:

- the constant $[\![\,]\!]$ (*empty multiset*), of sort bag;
- the operator $[\![\cdot]\!]^{(\cdot)}$ (*multiset construction*), of sort $\mathsf{ur} \times \mathsf{int} \to \mathsf{bag}$;
- the operators $\cup$ (*union*), $\uplus$ (*sum*), and $\cap$ (*intersection*), all of them having sort $\mathsf{bag} \times \mathsf{bag} \to \mathsf{bag}$;
- the operator $\mathrm{count}(\cdot,\cdot)$, of sort $\mathsf{ur} \times \mathsf{bag} \to \mathsf{int}$;

Finally, for each sort $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$, the language **BUI** contains an enumerable quantity of variables of sort τ and an equality symbol $=_\tau$ of sort $\tau \times \tau$.[3]

Definition 1. **BUI**-*terms (resp.* **BUI**-*formulae) are well-sorted terms (resp. formulae) constructed using the symbols of the language* **BUI**.

Definition 2. *A* bag-*term*[4] *is* PURE *if all symbols in it are either variables or one of* $[\![\,]\!]$, $[\![\cdot]\!]^{(\cdot)}$, $\cup$, $\uplus$, $\cap$. *Pure* ur-*terms and pure* int-*terms are defined similarly.*

A pure bag-*atom is an atom of the form* $s = t$, *where* s, t *are pure* bag-*terms. Similarly one can define pure* ur-*atom, pure* int-*atoms, and, in general, pure* τ-*formulae, for* $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$.

Unless otherwise specified, in the rest of the paper x, y, z will denote bag-variables, u, v, w will denote int-variables, and a, b, c will denote ur-variables.

2.3 The language BUI: semantics

Definition 3. *An* INTERPRETATION $\mathcal{A}$ *of* **BUI** *is a many-sorted interpretation of the sorts, variables and symbols in the language* **BUI** *satisfying the following conditions:*

- *each sort* $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$ *is mapped to a non-empty set* A_τ *such that:*
  - A_{ur} *is a non-empty set;*
  - A_{int} *is the set of all integers* $\mathbb{Z} = \{0, \pm 1, \pm 2, \dots\}$;
  - A_{bag} *is the collection of all multisets whose elements are in* A_{ur};
- *for each sort* τ, *each variable* x *of sort* τ *is mapped to an element* $x^{\mathcal{A}}$ *in* A_τ;
- *the symbols* $[\![\,]\!]$, $[\![\cdot]\!]^{(\cdot)}$, $\cup$, $\uplus$, $\cap$, count, 0, 1, $+$, $-$, max, min, $\leq$, *are interpreted according to their standard interpretation;*
- $=_\tau$ *is interpreted as the identity in* A_τ, *for* $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$.

In the rest of the paper the calligraphic letters $\mathcal{A}, \mathcal{B}, \dots$ will denote interpretations, and the corresponding Roman letters, properly subscripted, will denote the domains of the interpretations.

Definition 4. *A* **BUI**-*formula* φ *is*

- VALID, *if it evaluates to true in all interpretations;*
- SATISFIABLE, *if it evaluates to true in some interpretation;*
- UNSATISFIABLE, *if it evaluates to false in all interpretations.*

[3] We will write $=$ in place of $=_\tau$ when τ is clear from the context. In addition, we will use the notation $s \neq t$ as an abbreviation of $\neg(s = t)$.

[4] For a sort τ, a τ-term is a term of sort τ.

2.4 Theories

We use ur-*theories* in order to model the elements of sort ur.

Definition 5. *An* ur-THEORY *is any set of pure* ur-*formulae.*

Given an ur-theory T, a T-interpretation is an interpretation in which all formulae in T evaluate to true.

Definition 6. *Given an* ur-*theory* T*, a* **BUI**-*formula* φ *is*

- T-VALID*, if it evaluates to true in all* T*-interpretations;*
- T-SATISFIABLE*, if it evaluates to true in some* T*-interpretation;*
- T-UNSATISFIABLE*, if it evaluates to false in all* T*-interpretations.*

3 The decision procedure

Let T be an ur-theory for which a decision procedure for the T-satisfiability of quantifier-free pure ur-formulae is available. We now describe a decision procedure for checking the T-satisfiability of any quantifier-free **BUI**-formula φ. Note that, by converting φ into a disjunctive normal form, without loss of generality we may restrict ourselves to consider only conjunctions of **BUI**-literals.

The decision procedure consists of five phases, which we systematically describe in the next five subsections.

3.1 First phase: variable abstraction

The first phase of our decision procedure takes as input a conjunction φ of mixed **BUI**-literals, and converts it into a conjunction of pure **BUI**-literals. More specifically, the output of the variable abstraction phase is a pair $\langle \varphi', \varphi'' \rangle$ of conjunctions of **BUI**-literals with the following properties:

(a) $\varphi' \cup \varphi''$ is T-satisfiable if and only if so is φ;
(b) each literal in φ' is pure;
(c) each pure bag-literal in φ' is of the form

$$\begin{array}{cccc} x = y\,, & x \neq y\,, & x = [\![\,]\!]\,, & x = [\![a]\!]^{(u)}\,, \\ x = y \cup z\,, & x = y \uplus z\,, & x = y \cap z\,, & \end{array} \tag{1}$$

(d) each literal in φ'' is of the form $u = \mathrm{count}(a, x)$.

Note that all properties can be effectively enforced with the help of new auxiliary variables.

3.2 Second phase: partition

In the second phase we partition $\varphi' \cup \varphi''$ into four disjoint sets of literals φ_{ur}, φ_{int}, φ_{bag} and φ_{count} where

- φ_{ur} contains all pure ur-literals in φ';
- φ_{int} contains all pure int-literals in φ';
- φ_{bag} contains all pure bag-literals in φ';
- $\varphi_{\mathsf{count}} = \varphi''$.

We call $\varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ a conjunction of **BUI**-literals in *separate* form.

3.3 Third phase: variable generation

Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of **BUI**-literals in separate form. The third phase of our decision procedure does not change φ, but instead generates new ur-variables and new int-variables which will be used in the later phases.

More specifically, the variable generation phase consists of two steps:

Step 1: generate ur-variables. For each literal of the form $x \neq y$ in φ_{bag}, generate a new ur-variable $a_{x,y}$.

Step 2: generate int-variables. For each ur-variable a either generated in Step 1 or occurring in φ, and for each bag-variable x occurring in φ_{bag}, generate a new int-variable $w_{a,x}$.

Note that:

- the intuition behind Step 1 is that if two bags x, y are different then there must exist some element a such that $\mathrm{count}(a, x) \neq \mathrm{count}(a, y)$;
- the intuitive meaning of the variables generated in the second step is to represent, for each ur-element a and for each bag x, the value of $\mathrm{count}(a, x)$.

We denote with V_τ, for $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$, the collection of τ-variables that either occur in φ or are generated in the variable generation phase. We also denote with $wax(\varphi)$ the collection of literals $\{0 \leq w_{a,x} : a \in V_{\mathsf{ur}} \text{ and } x \in V_{\mathsf{bag}}\}$.

3.4 Fourth phase: decomposition

Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of **BUI**-literals in separate form. In the fourth phase of our decision procedure we nondeterministically guess an *arrangement* of φ.

Definition 7. *Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of* **BUI***-literals in separate form. An* ARRANGEMENT *of φ is any equivalence relation R on V_{ur}.*

For an arrangement R of φ, we define $res_{\mathsf{ur}}(\varphi, R)$ to be the following collection of literals

$$\begin{aligned} res_{\mathsf{ur}}(\varphi, R) = &\{a = b : a, b \in V_{\mathsf{ur}} \text{ and } aRb\} \ \cup \\ &\{a \neq b : a, b \in V_{\mathsf{ur}} \text{ and not } aRb\}\,, \end{aligned}$$

and we also define $res_{\mathsf{int}}(\varphi, R)$ to be the collection of literals obtained by replacing all literals in $\varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}} \cup res_{\mathsf{ur}}(\varphi, R)$ with the following formulae:[5]

$$\begin{array}{lcl}
x = y & \Longrightarrow & \bigwedge\limits_{a \in V_{\mathsf{ur}}} (w_{a,x} = w_{a,y}) \\
x \neq y & \Longrightarrow & w_{a_{x,y},x} \neq w_{a_{x,y},y} \\
x = [\![]\!] & \Longrightarrow & \bigwedge\limits_{a \in V_{\mathsf{ur}}} (w_{a,x} = 0) \\
x = [\![a]\!]^{(u)} & \Longrightarrow & (w_{a,x} = u) \wedge \bigwedge\limits_{\neg(bRa)} (w_{b,x} = 0) \\
x = y \cup z & \Longrightarrow & \bigwedge\limits_{a \in V_{\mathsf{ur}}} (w_{a,x} = \max(w_{a,y}, w_{a,z})) \\
x = y \uplus z & \Longrightarrow & \bigwedge\limits_{a \in V_{\mathsf{ur}}} (w_{a,x} = w_{a,y} + w_{a,z}) \\
x = y \cap z & \Longrightarrow & \bigwedge\limits_{a \in V_{\mathsf{ur}}} (w_{a,x} = \min(w_{a,y}, w_{a,z})) \\
u = \mathrm{count}(a, x) & \Longrightarrow & u = w_{a,x} \\
a = b & \Longrightarrow & \bigwedge\limits_{x \in V_{\mathsf{bag}}} (w_{a,x} = w_{b,x}) \\
a \neq b & \Longrightarrow & \mathit{true}\,.
\end{array}$$

All replacements are fairly intuitive, except maybe the last two ones. Indeed, for any multiset x, if $a = b$ then $\mathrm{count}(a, x) = \mathrm{count}(b, x)$. On the other hand, if $a \neq b$ then nothing can be said about $\mathrm{count}(a, x)$ and $\mathrm{count}(b, x)$.

Note also that the only replacement that needs to take into account nonequivalent **ur**-variables is the one dealing with the operator $[\![\cdot]\!]^{(\cdot)}$.

3.5 Fifth phase: check

Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of **BUI**-literals in separate form, and let R be the arrangement guessed in the decomposition phase.

The fifth and last phase of our decision procedure consists in:

1. checking $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ for T-satisfiability;
2. checking $\varphi_{\mathsf{int}} \cup \mathit{wax}(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$ for satisfiability in $\mathbb{Z}$.

[5] Where *true* is an abbreviation of $0 = 0$.

If both checks 1 and 2 succeed, we declare φ to be T-satisfiable.

Note that check 1 can be performed by using the decision procedure for the ur-theory T, whereas check 2 can be performed by using any decision procedure for integer linear arithmetic.[6]

3.6 An example

As an example of how our decision procedure works, let us consider the counting law for multisets:

$$x \uplus y = (x \cup y) \uplus (x \cap y)\,. \tag{2}$$

We want to show that (2) is T-valid, where T is the empty ur-theory. To do so, it is sufficient to prove that its negation

$$x \uplus y \neq (x \cup y) \uplus (x \cap y) \tag{3}$$

is T-unsatisfiable. After applying the variable abstraction and partition phases, we obtain the following conjunction $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ in separate form:

$$\varphi_{\mathsf{ur}} = \emptyset\,, \qquad \varphi_{\mathsf{int}} = \emptyset\,,$$

$$\varphi_{\mathsf{bag}} = \left\{ \begin{array}{l} z_0 = x \uplus y\,, \\ z_1 = x \cup y\,, \\ z_2 = x \cap y\,, \\ z_3 = z_1 \uplus z_2\,, \\ z_0 \neq z_3 \end{array} \right\}, \qquad \varphi_{\mathsf{count}} = \emptyset\,.$$

Since φ_{bag} contains only one disequality, in the variable generation phase we generate one new ur-variable a and six new int-variables $w_{a,x}$, $w_{a,y}$, w_{a,z_0}, w_{a,z_1}, w_{a,z_2}, w_{a,z_3}, obtaining the following conjunction $wax(\varphi)$:

$$wax(\varphi) = \{0 \leq w_{a,z} : z \in \{x, y, z_0, z_1, z_2, z_3, z_4\}\}\,.$$

In the decomposition phase, since $V_{\mathsf{ur}} = \{a\}$, the only permitted arrangement is $R = \{(a, a)\}$. We then have $res_{\mathsf{ur}}(\varphi, R) = \{a = a\}$ and

$$res_{\mathsf{int}}(\varphi, R) = \left\{ \begin{array}{l} w_{a,z_0} = w_{a,x} + w_{a,y} \\ w_{a,z_1} = \max(w_{a,x}, w_{a,y}) \\ w_{a,z_2} = \min(w_{a,x}, w_{a,y}) \\ w_{a,z_3} = w_{a,z_1} + w_{a,z_2} \\ w_{a,z_0} \neq w_{a,z_3} \end{array} \right\}.$$

Clearly, $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ is T-satisfiable. However, $\varphi_{\mathsf{int}} \cup wax(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$ has no solution in $\mathbb{Z}$, and therefore we conclude that (3) is T-unsatisfiable, hence (2) is T-valid.

[6] Since integer linear arithmetic is a decidable and complete theory.

4 Soundness, completeness, and decidability

In this section we prove that our decision procedure is sound and complete for the T-satisfiability of conjunctions of **BUI-**literals. Let us start with soundness.

Theorem 1 (soundness). *Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of* **BUI-***literals in separate form, and let T be an* ur*-theory. Assume that there exists an arrangement R of φ such that:*

(i) $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ is T-satisfiable;
(ii) $\varphi_{\mathsf{int}} \cup wax(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$ is satisfiable in $\mathbb{Z}$.

Then φ is T-satisfiable.

Proof. Let $\mathcal{A}$ be a T-interpretation satisfying $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ and let $\mathcal{B}$ be an interpretation satisfying $\varphi_{\mathsf{int}} \cup wax(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$.

We now define an interpretation $\mathcal{M}$. First, we specify the domains by letting $M_{\mathsf{ur}} = A_{\mathsf{ur}}$, $M_{\mathsf{int}} = \mathbb{Z}$ and $M_{\mathsf{bag}} = \{x : x \text{ is a multiset and } \mathrm{domain}(x) \subseteq A_{\mathsf{ur}}\}$. Then:

- for each ur-variable $a \in V_{\mathsf{ur}}$, we let $a^{\mathcal{M}} = a^{\mathcal{A}}$
- for each int-variable $u \in V_{\mathsf{int}}$, we let $u^{\mathcal{M}} = u^{\mathcal{B}}$.
- for each bag-variable $x \in V_{\mathsf{bag}}$, we let $x^{\mathcal{M}}$ be the unique multiset such that

$$\mathrm{domain}(x^{\mathcal{M}}) = \{a^{\mathcal{A}} : a \in V_{\mathsf{ur}} \text{ and } w^{\mathcal{B}}_{a,x} > 0\}\,,$$

and

$$\mathrm{count}(a^{\mathcal{A}}, x^{\mathcal{M}}) = w^{\mathcal{B}}_{a,x}\,, \qquad \text{for each } a \in V_{\mathsf{ur}}\,.$$

Note that in order for the above definition to be sound, we must ensure that for every two ur-variables a, b and for every bag-variable x, if $a^{\mathcal{A}} = b^{\mathcal{A}}$ then $w^{\mathcal{B}}_{a,x} = w^{\mathcal{B}}_{b,x}$. To see that this is the case, assume that $a^{\mathcal{A}} = b^{\mathcal{A}}$. But then the literal $a = b$ must occur in $res_{\mathsf{ur}}(\varphi, R)$, and therefore the literal $w_{a,x} = w_{b,x}$ is in $res_{\mathsf{int}}(\varphi, R)$. It follows $w^{\mathcal{B}}_{a,x} = w^{\mathcal{B}}_{b,x}$.

We claim that $\mathcal{M}$ is a T-interpretation satisfying φ. Clearly, $\mathcal{M}$ satisfies $T \cup \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}}$. Moreover, by inspecting the replacements in Subsection 3.4, it is easy to verify that $\mathcal{M}$ also satisfies all literals in $\varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$. □

Our decision procedure is also complete, as proved by the following theorem.

Theorem 2 (completeness). *Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a T-satisfiable conjunction of* **BUI-***literals in separate form, where T is an* ur*-theory.*

Then there exists an arrangement R of φ such that:

(a) $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ is T-satisfiable;
(b) $\varphi_{\mathsf{int}} \cup wax(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$ is satisfiable in $\mathbb{Z}$.

Proof. Let $\mathcal{M}$ be a T-interpretation satisfying φ. Let us define an arrangement R by putting:

$$uRv \quad \text{if and only if} \quad u^{\mathcal{M}} = v^{\mathcal{M}}\,, \qquad\qquad \text{for each } u, v \in V_{\mathsf{ur}}\,.$$

Clearly, property (a) holds since $\mathcal{M}$ satisfies $T \cup \varphi_{\mathsf{ur}}$ and, by construction, $\mathcal{M}$ also satisfies $res_{\mathsf{ur}}(\varphi, R)$.

Next, in order to verify property (b), notice that by assumption $\mathcal{M}$ satisfies φ_{int}. In addition, since the variables $w_{a,x}$'s do not occur in φ, we can safely modify $\mathcal{M}$ by letting $w_{a,x}^{\mathcal{M}} = \mathrm{count}(a^{\mathcal{M}}, x^{\mathcal{M}})$, for each $a \in V_{\mathsf{ur}}$ and $x \in V_{\mathsf{bag}}$. Clearly, by construction the modified $\mathcal{M}$ satisfies $wax(\varphi)$. Moreover, by inspecting the replacements in Subsection 3.4 it is easy to verify that the modified $\mathcal{M}$ also satisfies all literals in $res_{\mathsf{int}}(\varphi, R)$. □

Combining Theorems 1 and 2 with the observation that there is only a finite number of arrangements of any collection φ of **BUI**-literals in separate form, we obtain the following decidability result.

Theorem 3 (decidability). *Let T be an* ur*-theory for which a decision procedure for the T-satisfiability of quantifier-free pure* ur*-formulae is available. Then the T-satisfiability problem for quantifier-free formulae in the language* **BUI** *is decidable.*

5 NP-completeness

In this section we show that the T-satisfiability problem for quantifier-free **BUI**-formulae is $\mathcal{NP}$-complete, provided that the T-satisfiability problem for quantifier-free pure ur-formulae is in $\mathcal{NP}$.

Clearly, $\mathcal{NP}$-hardness follows by the fact that the propositional calculus is embedded into the language **BUI**. Thus, we obtain $\mathcal{NP}$-completeness if we show that the procedure described in Section 3 takes nondeterministic polynomial time, since guessing one of the disjuncts of a disjunctive normal form of any **BUI**-formula takes nondeterministic polynomial time.

Clearly, the variable abstraction and partition phases can be done in deterministic linear time in the size of the input formula.

Concerning the other phases, let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of **BUI**-literals in separate form. As usual, denote with V_τ, for $\tau \in \{\mathsf{bag}, \mathsf{ur}, \mathsf{int}\}$, the collection of τ-variables that either occur in φ or are generated in the variable generation phase.

Note that:

- the variable generation phase takes deterministic time $\mathcal{O}(t^2 + (s + t^2) \cdot t)$, where s is the number of ur-variables occurring in φ and t is the number of bag-variables occurring in φ;
- guessing an arrangement takes nondeterministic time $\mathcal{O}(|V_{\mathsf{ur}}|^2)$;
- computing $res_{\mathsf{ur}}(\varphi, R)$ takes deterministic time $\mathcal{O}(|V_{\mathsf{ur}}|^2)$.

- computing $res_{\mathsf{int}}(\varphi, R)$ takes deterministic time $\mathcal{O}(m \times (|V_{\mathsf{ur}}| + |V_{\mathsf{bag}}|))$, where m is the number of literals in $\varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}} \cup res_{\mathsf{ur}}(\varphi, R)$
- computing $wax(\varphi)$ takes deterministic time $\mathcal{O}(|V_{\mathsf{ur}}| \cdot |V_{\mathsf{bag}}|)$
- by assumption, checking the T-satisfiability of $\varphi_{\mathsf{ur}} \cup res_{\mathsf{ur}}(\varphi, R)$ take nondeterministic polynomial time.
- checking $\varphi_{\mathsf{int}} \cup wax(\varphi) \cup res_{\mathsf{int}}(\varphi, R)$ for satisfiability in $\mathbb{Z}$ takes nondeterministic polynomial time (cf. [2]).

Thus, we obtain the following $\mathcal{NP}$-completeness result.

Theorem 4 ($\mathcal{NP}$-completeness). *Let T be an* ur*-theory such that the T-satisfiability problem for quantifier-free pure* ur*-formulae is in $\mathcal{NP}$.*

Then the T-satisfiability problem for quantifier-free formulae in the language **BUI** *is $\mathcal{NP}$-complete.*

6 Efficiency issues

Since the language **BUI** properly extends both the propositional calculus and integer linear arithmetic, the $\mathcal{NP}$-completeness result proved in the previous section is the best complexity result one could expect from a theoretical point of view. From a pragmatic point of view, however, the decision procedure described in Section 3 is not susceptible of a practical and efficient implementation because of a very expensive nondeterministic decomposition phase. Nevertheless, in this section we show that, under some additional assumptions, the nondeterminism induced by the decomposition phase can be avoided by appealing to the simple idea of *convexity*.

The notion of convexity was introduced in [1] in order to avoid case splitting and make efficient the deterministic version of the Nelson-Oppen combination method. The following is a formal definition.

Definition 8. *An* ur*-theory T is* CONVEX *if for every collection φ of pure* ur*-literals and for every disjunction of pure* ur*-equalities $\bigvee_{i=1}^{n} s_i = t_i$, if $T \cup \varphi \models \bigvee_{i=1}^{n} s_i = t_i$ then $T \cup \varphi \models s_i = t_i$, for some $i \in \{1, \ldots, n\}$.*

Unfortunately, the language **BUI** is not convex. In fact:

- the theory of integer linear arithmetic is not convex. As an example, the conjunction $1 \leq u \leq 2 \wedge v = 1 \wedge w = 2$ entails the disjunction $u = v \vee u = w$ but does not entail neither $u = v$ nor $u = w$;
- the underlying theory T of the ur-elements is arbitrary, and therefore it may not be convex;
- the operator $[\![\cdot]\!]^{(\cdot)}$ is also a source of non-convexity. For example, the literal $a \in [\![b]\!]^{(1)} \uplus [\![c]\!]^{(1)}$ entails $a = b \vee a = c$ but does not entail neither $a = b$ nor $a = c$.

The non convexity of integer linear arithmetic is not a problem since the decomposition phase involves only ur-variables and not int-variables. Thus, in order to obtain an efficient deterministic version of our decision procedure for

BUI, all is needed is to assume that T is convex and that there is no occurrence of the operator $[\![\cdot]\!]^{(\cdot)}$ in the formula to be checked for T-satisfiability.

In this deterministic procedure, the variable abstraction, partition, and variable generation phases are exactly as the ones described in Section 3. The decomposition phase is, however, replaced by the following equality propagation phase.

> **Equality propagation.** Let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be the conjunction of **BUI**-literals in separate form obtained in the partition phase. If φ_{ur} is T-unsatisfiable, declare φ to be T-unsatisfiable. Otherwise, compute the collection of literals $\mathit{det\text{-}res}_{\mathsf{ur}}(\varphi)$ and $\mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$ defined as follows:
> - $\mathit{det\text{-}res}_{\mathsf{ur}}(\varphi) = \{a = b : a, b \in V_{\mathsf{ur}} \text{ and } T \cup \varphi_{\mathsf{ur}} \models a = b\}$;[7]
> - $\mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$ is the collection of literals obtained by applying the replacements in Subsection 3.4 to $\varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}} \cup \mathit{det\text{-}res}_{\mathsf{ur}}(\varphi)$.

After applying the equality propagation phase, we check whether $\varphi_{\mathsf{int}} \cup \mathit{wax}(\varphi) \cup \mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$ is satisfiable in $\mathbb{Z}$. If this is the case, we declare φ to be T-satisfiable, otherwise we declare φ to be T-unsatisfiable.

Note that the deterministic procedure just described is much more efficient than the one described in Section 3. Efficiency mainly stems from the fact that equalities are propagated only in one direction, with the effect that only *one* integer linear system needs to be solved, whereas in the procedure described in Section 3 we solve a linear system for each guessed arrangement.

Before proving that the deterministic procedure presented in this section is correct, we need the following technical lemma.

Lemma 1. *Let V be a collection of* ur*-variables, and let φ be a T-satisfiable conjunction of* ur*-literals. Then there exists a T-interpretation $\mathcal{A}$ satisfying φ such that $a^{\mathcal{A}} \neq b^{\mathcal{A}}$, for all* ur*-variables $a, b \in V$ for which $T \cup \varphi \not\models a = b$.*

Proof. Let $S = \{(a, b) : a, b \in V \text{ and } T \cup \varphi \not\models a = b\}$, and consider the disjunction

$$\psi : \bigvee_{(a,b) \in S} a = b\,.$$

If $T \cup \varphi \not\models \psi$ then the lemma is proved. If instead $T \cup \varphi \models \psi$ then, by the convexity of T, there exists a pair $(a, b) \in S$ such that $T \cup \varphi \models a = b$, a contradiction. □

The following two theorems show the correctness of the deterministic procedure presented in this section.

Theorem 5 (soundness). *Let T be a convex* ur*-theory and let $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ be a conjunction of* **BUI***-literals in separate form not involving the operator $[\![\cdot]\!]^{(\cdot)}$. Assume that:*

[7] Note that $\mathit{det\text{-}res}_{\mathsf{ur}}(\varphi)$ can be effectively computed due to our decidability assumptions on T.

(i) φ_{ur} *is* T*-satisfiable;*
(ii) $\varphi_{\mathsf{int}} \cup \mathit{wax}(\varphi) \cup \mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$ *is satisfiable in* $\mathbb{Z}$*.*

Then φ *is* T*-satisfiable*

Proof. Since T is convex, by Lemma 1 there exists a T-interpretation $\mathcal{A}$ satisfying φ_{ur} such that $a^{\mathcal{A}} \neq b^{\mathcal{A}}$, for all ur-variables $a, b \in V_{\mathsf{ur}}$ for which $T \cup \varphi_{\mathsf{ur}} \not\models a = b$. Let also $\mathcal{B}$ be an interpretation satisfying $\varphi_{\mathsf{int}} \cup \mathit{wax}(\varphi) \cup \mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$.

We now define an interpretation $\mathcal{M}$ exactly as it was done in the proof of Theorem 1. As before, in order to ensure that $\mathcal{M}$ is well-defined, we need to ensure that for every two ur-variables a, b and for every bag-variable x, if $a^{\mathcal{A}} = b^{\mathcal{A}}$ then $w^{\mathcal{B}}_{a,x} = w^{\mathcal{B}}_{b,x}$.

Thus, assume that $a^{\mathcal{A}} = b^{\mathcal{A}}$. Then $T \cup \varphi_{\mathsf{ur}} \models a = b$ and therefore the literal $a = b$ must occur in $\mathit{res}_{\mathsf{ur}}(R)$. Thus, the literal $w_{a,x} = w_{b,x}$ is in $\mathit{res}_{\mathsf{int}}(\varphi, R)$, which implies that $w^{\mathcal{B}}_{a,x} = w^{\mathcal{B}}_{b,x}$.

The proof can now continue smoothly by verifying, exactly as done in the proof of Theorem 1, that $\mathcal{M}$ is a T-interpretation satisfying φ. □

Theorem 6 (completeness). *Let* T *be an* ur*-theory, and let* $\varphi = \varphi_{\mathsf{ur}} \cup \varphi_{\mathsf{int}} \cup \varphi_{\mathsf{bag}} \cup \varphi_{\mathsf{count}}$ *be a* T*-satisfiable conjunction of* **BUI***-literals in separate form not involving the operator* $[\![\cdot]\!]^{(\cdot)}$*.*

Then the following holds:

(a) φ_{ur} *is* T*-satisfiable;*
(b) $\varphi_{\mathsf{int}} \cup \mathit{wax}(\varphi) \cup \mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$ *is satisfiable in* $\mathbb{Z}$*.*

Proof. Obviously φ_{ur} is T-satisfiable. In addition, since the variables $w_{a,x}$'s do not occur in φ, we can safely modify $\mathcal{M}$ by letting $w^{\mathcal{M}}_{a,x} = \mathrm{count}(a^{\mathcal{M}}, x^{\mathcal{M}})$, for each $a \in V_{\mathsf{ur}}$ and $x \in V_{\mathsf{bag}}$. Clearly, by construction the modified $\mathcal{M}$ satisfies $\mathit{wax}(\varphi)$. Moreover, by inspecting the replacements in Subsection 3.4 it is easy to verify that the modified $\mathcal{M}$ also satisfies all literals in $\mathit{det\text{-}res}_{\mathsf{int}}(\varphi)$. □

7 Conclusion

We defined the constraint language **BUI** for combining multisets, integers, and ur-elements. We then presented a decision procedure for **BUI** and we proved its correctness. In particular, we showed that our decision procedure remains correct even if the underlying theory T of the ur-elements is not stably infinite.

We also addressed the problem of efficiency by presenting a second procedure which avoids the expensive nondeterminism of the decomposition phase. The second procedure is correct provided that T is convex and that the formula to be checked for T-satisfiability does not involve the operator $[\![\cdot]\!]^{(\cdot)}$.

Finally, we proved that if the T-satisfiability problem for quantifier-free pure ur-formulae is in $\mathcal{NP}$ then the T-satisfiability problem for quantifier-free **BUI**-formulae is $\mathcal{NP}$-complete.

Although not shown in this paper, it is easy to verify that the results of Sections 4 and 5 generalize to the case of multisets of integers, thus allowing the

expression of constraints such as $\mathrm{count}(\mathrm{count}(a, x), x) = 1$, which are forbidden by the syntax of **BUI**. However, due to the non convexity of integer linear arithmetic, it is not clear to us whether the same generalization can be done for the results of Section 6.

Acknowledgments

We thank Cesare Tinelli and the anonymous reviewers for useful comments.

This research was supported in part by NSF(ITR) grant CCR-01-21403, by NSF grant CCR-99-00984-001, by ARO grant DAAD19-01-1-0723, and by ARPA/AF contracts F33615-00-C-1693 and F33615-99-C-3014.

References

1. Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, 1979.
2. Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the Association for Computing Machinery*, 28(4):765–768, 1981.
3. Cesare Tinelli and Mehdi T. Harandi. A new correctness proof of the Nelson-Oppen combination procedure. In Franz Baader and Klaus U. Schulz, editors, *Frontiers of Combining Systems*, volume 3 of *Applied Logic Series*, pages 103–120. Kluwer Academic Publishers, 1996.
4. Calogero G. Zarba. Combining lists with integers. In Rajeev Goré, Alexander Leitsch, and Tobias Nipkow, editors, *International Joint Conference on Automated Reasoning (Short Papers)*, Technical Report DII 11/01, pages 170–179. University of Siena, Italy, 2001.
5. Calogero G. Zarba. Combining sets with integers. In Alessandro Armando, editor, *Frontiers of Combining Systems*, volume 2309 of *Lecture Notes in Artificial Intelligence*, pages 103–116. Springer, 2002.