A new Presburger arithmetic
decision procedure
based on extended Prolog execution

Laurent Fribourg
L.I.E.N.S. - 45 rue d’Ulm, 75005 Paris - France
fribourg@dmi.ens.fr

Abstract

In this paper we are concerned by the problem of determining validity of universal Presburger formulas. The original point here is that we do not use a specific algorithm but attempt to prove Presburger formulas by induction using Kanamori et al.’ verification system of extended Prolog execution. This leads us to a new decision algorithm for which a proof of correctness is given.

1 Introduction

In this paper we are interested by the decision problem for Presburger formulas. A Presburger formula is a first-order formula allowing only natural numbers built over 0, s(uccessor) and +, variables, inequality relations, disjunction and conjunction connectives and quantification. Presburger arithmetic has been used in the field of proving assertions about programs [Bledsoe][Suzuki & Jefferson]. It is a decidable theory, and several methods can be found in the literature [Cooper][Bledsoe][Shostak]. These methods are very specific, and we would like to know if a general purpose theorem-prover would be capable to solve the decision problem. The prover that we will consider is Kanamori et al.’ system of “extended Prolog execution”. The system makes heavy use of the rule of computation induction [Clark] which allows to treat universally quantified formulas. We show how this verification system can be used to determine the validity of universal Presburger formulas (i.e. formulas without existential quantifier). This yields a new algorithm whose correctness is proven.
2 Preliminaries

The Presburger formulas that we consider have the following form
\[ \forall x \ t^1(x) \leq u^1(x) \lor \ldots \lor t^n(x) \leq u^n(x). \]
where \( x \) denotes a vector of \( m \) variables, and \( t^1(x), u^1(x), \ldots, t^n(x), u^n(x) \) denote arithmetical terms built on 0, s and +. (For the sake of simplicity, we will not introduce the conjunction connective but the method works as well.)

Such a formula will actually be represented under the form
\[ \forall x \exists \alpha_1, \ldots, \exists \alpha_n \ le(\alpha^1, t^1(x), u^1(x)) \land \ldots \land le(\alpha^n, t^n(x), u^n(x)) \land or_n(\alpha^1, \ldots, \alpha^n) \]
where \( \alpha^1, \ldots, \alpha^n \) denote boolean variables, \( le \) denotes the less-than-or-equal-to predicate, and \( or_n \) the \( n \)-ary disjunctive relation.

Following [Kanamori & Seki] we drop the quantifiers: the variables of the vector \( x \) become free variables while the existential boolean variables \( \alpha^i \) are preceded by the symbol ‘\(?\)’. We get
\[ le(\?\alpha^1, t^1(x), u^1(x)) \land \ldots \land le(\?\alpha^n, t^n(x), u^n(x)) \land or_n(\?\alpha^1, \ldots, \?\alpha^n). \]
In the examples, for the sake of readability, we will write natural numbers under their decimal form (e.g., 3 for \( s(s(s(0))) \)) and make use of multiplication by a constant (e.g., \( 3x \) for \( x + x + x \)).

3 Principle of the Procedure

Let us illustrate the principle of our procedure on an example (borrowed from [Bledsoe]). Consider the Presburger formula
\[ G : \ le(\alpha^1, 11, 5x) \land le(\alpha^2, 7x, 15) \land or_2(\alpha^1, \alpha^2). \]

1. By induction, we get on the one hand
\[ G_0 : \ le(\alpha^1, 11, 5 \times 0) \land le(\alpha^2, 7 \times 0, 15) \land or_2(\alpha^1, \alpha^2). \]
which does not contain \( x \) any longer, and on the other hand
\[ H : \ le(\alpha^1, 11, 5(x + 1)) \land le(\alpha^2, 7(x + 1), 15) \land or_2(\alpha^1, \alpha^2) \]
\[ \iff le(\beta^1, 11, 5x) \land le(\beta^2, 7x, 15) \land or_2(\beta^1, \beta^2). \]

2. The formula \( H \) rewrites as
\[ I : \ le(\alpha^1, 11, 5x + 5) \land le(\alpha^2, 7x + 7, 15) \land or_2(\alpha^1, \alpha^2) \]
\[ \iff le(\beta^1, 11, 5x) \land le(\beta^2, 7x, 15) \land or_2(\beta^1, \beta^2). \]

3. Since the hypothesis atom \( or_2(\beta^1, \beta^2) \) holds iff \( \beta^1 \) or \( \beta^2 \) is true, the formula \( I \) may be decomposed into
\[ K_1 : \ le(\alpha^1, 11, 5x + 5) \land le(\alpha^2, 7x + 7, 15) \land or_2(\alpha^1, \alpha^2) \]
\[ \iff le(true, 11, 5x) \land le(\beta^2, 7x, 15). \]
\[ K_2 : \ le(\alpha^1, 11, 5x + 5) \land le(\alpha^2, 7x + 7, 15) \land or_2(\alpha^1, \alpha^2) \]
\[ \iff le(\beta^1, 11, 5x) \land le(true, 7x, 15). \]
By substituting \( ?\alpha^1 \) by \( true \) in \( K_1 \), and eliminating the first hypothesis and first conclusion atoms, we simplify \( K_1 \) into the trivial form.