Size-Change Termination Analysis in k-Bits

Michael Codish[1*], Vitaly Lagoon[2], Peter Schachte[2], and Peter J. Stuckey[2,3]

[1] Department of Computer Science, Ben-Gurion University, Israel
[2] Department of Computer Science and Software Engineering
The University of Melbourne, Australia
[3] NICTA Victoria Laboratory
mcodish@cs.bgu.ac.il, {lagoon,schachte,pjs}@cs.mu.oz.au

Abstract. Size-change termination analysis is a simple and powerful technique successfully applied for a variety of programming paradigms. A main advantage is that termination for size-change graphs is decidable and based on simple linear ranking functions. A main disadvantage is that the size-change termination problem is PSPACE-complete. Proving size change termination may have to consider exponentially many size change graphs. This paper is concerned with the representation of large sets of size-change graphs. The approach is constraint based and the novelty is that sets of size-change graphs are represented as disjunctions of size-change constraints. A constraint solver to facilitate size-change termination analysis is obtained by interpreting size-change constraints over a sufficiently large but finite non-negative integer domain. A Boolean k-bit modeling of size change graphs using binary decision diagrams leads to a concise representation. Experimental evaluation indicates that the 2-bit representation facilitates an efficient implementation which is guaranteed complete for our entire benchmark suite.

1 Introduction

Size-change termination analysis [8] is a simple and powerful technique to verify program termination. First, the transition relation of a program is approximated by a set of size-change graphs. Then, termination is guaranteed if all of the idempotent size change graphs in the closure of this set under a composition operation have (possibly different) ranking functions.

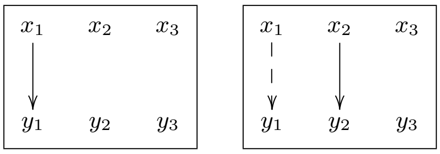
A typical example is the analysis of the Prolog program depicted in Figure 1**(a)** which computes Ackermann's function. The size-change graphs in the figure describe all transitions in computations of this program. Between subsequent function calls, either the first argument decreases in size (Figure 1**(b)**), or else it does not increase and the second argument decreases in size (Figure 1**(c)**). As formalised below, these graphs are idempotent, closed under composition and have as ranking functions $f(\bar{u}) = u_1$ and $f'(\bar{u}) = u_2$ respectively.

A major strength of the technique is that for a given set of size-change graphs termination is decidable. An idempotent size-change graph has a ranking function if and only if it has one which indicates that a specific single argument

* Research performed at the University of Melbourne

$ackerman(0, N, s(N)).$
$ackerman(s(M), 0, Res) \leftarrow$
$\quad ackerman(M, s(0), Res).$
$ackerman(s(M), s(N), Res) \leftarrow$
$\quad ackerman(s(M), N, Res1),$
$\quad ackerman(M, Res1, Res).$

(a)



(b) **(c)**

Fig. 1. Ackermann's function with size-change graphs

decreases in size. A major weakness is that size change termination is complete for PSPACE. While in practice this rarely occurs, closure under composition may introduce exponentially many additional size-change graphs.

This paper is concerned with the representation of large sets of size-change graphs and supporting operations for closing these representations under composition and testing all graphs in the closure for the existence of ranking functions. The key idea in our approach is to view sets of size-change graphs as constraints. For individual size-change graphs this idea is not new. The TerminWeb [9] analyser maintains sets of size-change graphs, each graph represented as a conjunction of constraints. The novelty in this paper is to illustrate how sets of size-change graphs can be represented accurately through disjunction. For example the two graphs in Figure 1 are captured by the constraint $(x_1 > y_1) \vee (x_1 \geq y_1 \wedge x_2 > y_2)$. Given this view, a set of size change graphs is equivalent to its set of solutions over the domain of non-negative integers, much the same as a Boolean function is equivalent to its set of models. We draw on the motivation that representing large sets of models for Boolean functions is a well studied problem with readily available off-the-shelf tools. The main difficulty is to provide set-based operations for size change termination which operate accurately on these representations.

To support an operation to compose disjunctions of size-change graphs we introduce a non-standard interpretation of the binary size-relations $>$ and $\geq$. This enables us to model composition of sets of size-change graphs as conjunction. To determine if each of the graphs in a set has a ranking function we apply a previous result [3] to design a suitable test.

Another difficulty is to provide a constraint solver for size-change graphs and their operations. This is achieved by interpreting constraints over a sufficiently large but finite domain (of non-negative integers). Finite domain constraints are then represented as Boolean functions as proposed in [6]. This Boolean representation for finite domain constraints and operations leads to an efficient implementation using binary decision diagrams. Experimental evaluation indicates that the 2-bit representation is guaranteed complete for our entire extensive benchmark suite. Of course the approach we describe does not ameliorate the PSPACE hardness of the termination problem for size change graphs, the resulting binary decision diagrams can require exponential space and time.

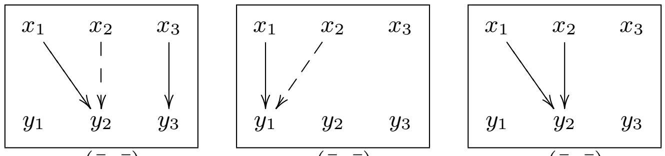


Fig. 2. Size-change graphs

2 Size-change termination

This section presents the standard definitions and results for size change graphs. Our definitions are similar to those given in [8] except that they are given in a language of constraints. The constraint representation naturally provides a notion of ordering not present in the original definition [8].

Definition 1 (size-change graphs - I). *A size-change graph is a binary clause of the form $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y})$ where $\bar{x}$ and $\bar{y}$ are the disjoint vectors of arguments and $c(\bar{x}, \bar{y})$ is a conjunction of constraints of the form $x \succ^b y$ with $x \in \bar{x}$, $y \in \bar{y}$ and $b \in \{0, 1\}$. A constraint $x \succ^b y$ corresponds to an edge and is interpreted as $x \geq y + b$: strict ($x > y$) or non-strict ($x \geq y$) when respectively $b = 1$ or $b = 0$.*

Size-change graph notation: Consider a size-change graph $g = p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y})$ with $\bar{x} = \langle x_1 \ldots, x_n \rangle$ and $\bar{y} = \langle y_1 \ldots, y_m \rangle$. We sometimes write g in the form $p_{/n}(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q_{/m}(\bar{y})$ to make explicit the arities of $\bar{x}$ and $\bar{y}$. The parameter set of g, is denoted $Par(g) = \{p_{\langle 1 \rangle}, \ldots, p_{\langle n \rangle}, q_{\langle 1 \rangle}, \ldots, q_{\langle m \rangle}\}$. For a set of size-change graphs G, we denote $Par(G) = \cup \left\{ Par(g) \,\middle|\, g \in G \right\}$. A size change graph of the form $p_{/n}(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p_{/n}(\bar{y})$ is called a *recursive* size change graph. When p and q are clear from the context we refer to $g = c(\bar{x}, \bar{y})$ as the size-change graph. In the examples, edges are depicted by solid and dashed arrows corresponding to strict and non-strict edges. For each pair of nodes $x \in \bar{x}$ and $y \in \bar{y}$ the unique strictest constraint between x and y is depicted.

Example 1. The following size-change graphs are depicted in Figure 2 as $c_1(\bar{x}, \bar{y})$, $c_2(\bar{x}, \bar{y})$ and $c_3(\bar{x}, \bar{y})$ respectively.

$$g_1 = \ p(x_1, x_2, x_3) \leftarrow x_1 > y_2, x_2 \geq y_2, x_3 > y_3, p(y_1, y_2, y_3).$$
$$g_2 = \ p(x_1, x_2, x_3) \leftarrow x_1 > y_1, x_2 \geq y_1, p(y_1, y_2, y_3).$$
$$g_3 = \ p(x_1, x_2, x_3) \leftarrow x_1 > y_2, x_2 > y_2, p(y_1, y_2, y_3).$$

Note that by Definition 1 the size-change graph

$$g_2' = \ p(x_1, x_2, x_3) \leftarrow x_1 > y_1, x_1 \geq y_1, x_2 \geq y_1, p(y_1, y_2, y_3).$$

is also depicted as $c_2(\bar{x}, \bar{y})$.

Definition 2 (size-change graph solution). *A solution θ for a size-change graph $c(\bar{x}, \bar{y})$ is a valuation on the variables $\bar{x}$ and $\bar{y}$, $\theta = \{x_1/a_1, \ldots, x_n/a_n, y_1/b_1, \ldots, y_m/b_m\}$ which is a solution of $c(\bar{x}, \bar{y})$, i.e. $c(\bar{a}, \bar{b})$ is valid.*

Solutions can be written as two rows $(\bar{a}, \bar{b})$ in a matrix, as illustrated in the following example.

Example 2. Consider the following 8 solutions and the size-change graphs $c_1(\bar{x}, \bar{y})$, $c_2(\bar{x}, \bar{y})$ and $c_3(\bar{x}, \bar{y})$ of Figure 2.

$$s_1 = \begin{bmatrix} 8,7,3 \\ 9,7,2 \end{bmatrix} \qquad s_2 = \begin{bmatrix} 4,3,8 \\ 3,7,9 \end{bmatrix} \qquad s_3 = \begin{bmatrix} 8,7,2 \\ 9,6,3 \end{bmatrix} \qquad s_4 = \begin{bmatrix} 8,7,2 \\ 5,6,3 \end{bmatrix}$$
$$s'_1 = \begin{bmatrix} 1,0,1 \\ 1,0,0 \end{bmatrix} \qquad s'_2 = \begin{bmatrix} 1,0,0 \\ 0,1,1 \end{bmatrix} \qquad s'_3 = \begin{bmatrix} 1,1,0 \\ 1,0,1 \end{bmatrix} \qquad s'_4 = \begin{bmatrix} 1,1,1 \\ 0,0,0 \end{bmatrix}$$

s_1 and s'_1 are solutions only for $c_1(\bar{x}, \bar{y})$, s_2 and s'_2 are solutions only for $c_2(\bar{x}, \bar{y})$, s_3 and s'_3 are solutions only for $c_3(\bar{x}, \bar{y})$, s_4 is a solution for $c_2(\bar{x}, \bar{y})$ and $c_3(\bar{x}, \bar{y})$ but not for $c_1(\bar{x}, \bar{y})$ and s'_4 is a solution for all three of the size-change graphs.

Definition 3 (order on size-change graphs). *Size-change graphs on the same parameter set are ordered by constraint entailment. A size change graph $c_1(\bar{x}, \bar{y})$ is more general than $c_2(\bar{x}, \bar{y})$ if the solutions of c_1 are a superset of the solutions of c_2, i.e., $c_2(\bar{x}, \bar{y}) \models c_1(\bar{x}, \bar{y})$. Size-change graphs are equivalent if they have the same sets of solutions, i.e. $c_1(\bar{x}, \bar{y}) \leftrightarrow c_2(\bar{x}, \bar{y})$.*

Definition 4 (composition and idempotence of size-change graphs). *Let $p(\bar{x}) \leftarrow c_1(\bar{x}, \bar{y}), q(\bar{y})$ and $q(\bar{x}) \leftarrow c_2(\bar{x}, \bar{y}), r(\bar{y})$ be size-change graphs. Their composition is the size-change graph $p(\bar{x}) \leftarrow c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}), r(\bar{y})$ given by*

$$c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}) = \bigwedge \left\{ x \succ^b y \;\middle|\; \begin{array}{l} x \in \bar{x},\ y \in \bar{y}, \\ c_1(\bar{x}, \bar{z}) \wedge c_2(\bar{z}, \bar{y}) \models x \succ^b y \end{array} \right\}.$$

Recursive size-change graph $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p(\bar{y})$ is idempotent *if and only if $c(\bar{x}, \bar{y}) \circ c(\bar{x}, \bar{y}) = c(\bar{x}, \bar{y})$. The pairwise composition of sets of size-change graphs G_1 and G_2, respectively of the form $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y})$ and $q(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), r(\bar{y})$ is: $G_1 \circ G_2 = \left\{ g_1 \circ g_2 \mid g_1 \in G_1, g_2 \in G_2 \right\}$.*

Definition 5 (closure under composition). *Let G be a set of size-change graphs. We denote by G^* the closure of G under composition. This is the smallest superset of G such that if $p(\bar{x}) \leftarrow c_1(\bar{x}, \bar{y}), q(\bar{y}) \in G^*$ and $q(\bar{x}) \leftarrow c_2(\bar{x}, \bar{y}), r(\bar{y}) \in G^*$ then also $p(\bar{x}) \leftarrow c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}), r(\bar{y}) \in G^*$.*

Example 3. The set of size-change graphs depicted in Figure 1 is closed under composition. Both graphs are idempotent. The graphs in Figure 2 are also idempotent. The graphs in Figure 3 are not idempotent.

Lee *et al.*[8] introduce the property of *size change termination* and prove that a set of size-change graphs G has this property if and only if each idempotent size-change graph $p_{/n}(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p_{/n}(\bar{y})$ in G^* has a strict "vertical down arrow" of the form $x_i > y_i$. Each individual idempotent graph has a strict vertical down arrow if and only if the following condition holds:

$$\bigvee_{i=1}^{n} \left(c(\bar{x}, \bar{y}) \models x_i > y_i \right). \tag{1}$$

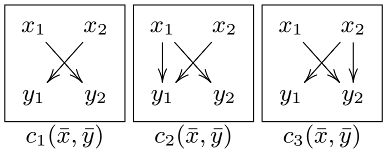


Fig. 3. Non-idempotent size change graphs.

For any (recursive) size change graph $c(\bar{x}, \bar{y})$, a function f mapping tuples of non-negative integers to a well founded domain and such that $c(\bar{x}, \bar{y}) \models f(\bar{x}) > f(\bar{y})$ is called a ranking function for $c(\bar{x}, \bar{y})$. Equation 1 implies that $c(\bar{x}, \bar{y})$ has a ranking function of the form $f(u_1, \ldots, u_n) = u_i$.

The result of [8] is generalized in [3] where the authors show that a set of size-change graphs G satisfies size-change termination if and only if the following condition holds: for every recursive (not necessarily idempotent) size-change graph $p_{/n}(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p_{/n}(\bar{y})$ in G^*, each solution of $c(\bar{x}, \bar{y})$ has a strict "vertical down arrow" of the form $x_i > y_i$. In other words, each individual recursive $c(\bar{x}, \bar{y})$ in G^* must satisfy the condition below:

$$c(\bar{x}, \bar{y}) \models \bigvee_{i=1}^{n} (x_i > y_i). \tag{2}$$

Equation 2 implies that $c(\bar{x}, \bar{y})$ has a ranking function of the form $f(u_1, \ldots, u_n) = \Sigma a_i u_i$ with all coefficients $a_i \in \{0, 1\}$. The distinction between the tests in Equations 1 and 2 is exemplified by the size change graph $c_1(\bar{x}, \bar{y})$ of Figure 3 which is not idempotent and which has no strict vertical down arrow of the form $x_i > y_i$. However, any solution of $c_1(\bar{x}, \bar{y})$ is also a solution of $c_2(\bar{x}, \bar{y})$ or of $c_3(\bar{x}, \bar{y})$ (in the same figure) which do have vertical down arrows. Note that the function $f(\bar{u}) = u_1 + u_2$ is a ranking function for $c_1(\bar{x}, \bar{y})$. Sidestepping the restriction to idempotent graphs turns out to be important to facilitate the specification of a set-based test for termination given in Section 3. In the following we refer to the implicant in Equation 2 as the *ranking constraint*.

Definition 6 (size-change ranking constraint). *Let $p_{/n}(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p_{/n}(\bar{y})$ be a recursive size-change graph. The corresponding ranking constraint is denoted*

$$\mathcal{R}(\bar{x}, \bar{y}) = \bigvee_{i=1}^{n} x_i \succ^1 y_i.$$

The application of size-change termination to proving termination is based on the observation that if a set of size-change graphs G is a safe approximation of the transition relation for a program P, and G satisfies size-change termination, then P terminates.

3 Set based size-change termination

In this section we propose a set based approach to size-change termination. The basic idea is that sets of size-change graphs can be represented as disjunctions of constraints with no loss of information for termination analysis. The contribution is in the design of the set-based operations for size-change termination analysis. The following definition provides the basic representation for a set of size-change graphs as a disjunction of constraints.

Definition 7 (disjunctive representation). *Let $G_{p,q}$ be a set of size-change graphs of the form $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y})$ (p and q are fixed). The disjunctive representation of $G_{p,q}$ is the binary clause denoted $G^{\vee}_{p,q} = p(\bar{x}) \leftarrow C(\bar{x}, \bar{y}), q(\bar{y})$ where $C(\bar{x}, \bar{y}) = \vee \left\{ c(\bar{x}, \bar{y}) \,\middle|\, p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y}) \in G_{p,q} \right\}$. When clear from the context we refer to $C(\bar{x}, \bar{y})$ as the disjunctive representation.*

This definition is easily extended to apply to sets of graphs with different source and target (p and q). In this case the result is a set of disjunctive constraints, (at most) one for each p and q.

Definition 8 (order on disjunctive size-change graphs). *Disjunctive size-change graphs with the same source and target are ordered by entailment. A disjunctive size-change graph G_1 is more general than G_2 if the solutions of G_1 include those of G_2. Two disjunctive size-change graphs are equivalent if they have the same sets of solutions.*

Example 4. Consider the size-change graphs $p(\bar{x}) \leftarrow c_i(\bar{x}, \bar{y}), p(\bar{y})$ for $i \in \{1, 2, 3\}$ depicted in Figure 3. The sets of graphs $\{c_1(\bar{x}, \bar{y})\}$ and $\{c_2(\bar{x}, \bar{y}), c_3(\bar{x}, \bar{y})\}$ are equivalent. In one direction, graph $c_1(\bar{x}, \bar{y})$ is more general than each of the graphs $c_2(\bar{x}, \bar{y})$ and $c_3(\bar{x}, \bar{y})$ which have fewer solutions (more constraints). In the other direction, observe that $c_1(\bar{x}, \bar{y}) \models x_1 + x_2 > y_1 + y_2 \models x_1 > y_1 \vee x_2 > y_2$ and so any solution of $c_1(\bar{x}, \bar{y})$ is either a solution of $c_2(\bar{x}, \bar{y})$ or of $c_3(\bar{x}, \bar{y})$.

When composing disjunctions of constraints we can no longer consider the original disjuncts as these are not maintained as sets. However we may consider all disjuncts that entail a given constraint. The following definition is intended only as the specification of set-based composition. We do not propose to implement the operation based on this definition. That would be very inefficient.

Definition 9 (composing disjunctive representations). *Let $G_{p,q}$ and $G_{q,r}$ be sets of size-change graphs with disjunctive representations $G^{\vee}_{p,q} = p(\bar{x}) \leftarrow C_1(\bar{x}, \bar{y}), q(\bar{y})$ and $G^{\vee}_{q,r} = q(\bar{x}) \leftarrow C_2(\bar{x}, \bar{y}), r(\bar{y})$ respectively. Their disjunctive composition is the size-change graph $G^{\vee}_{p,q} \circ G^{\vee}_{q,r} = p(\bar{x}) \leftarrow C(\bar{x}, \bar{y}), r(\bar{y})$ where*

$$C(\bar{x}, \bar{y}) = \bigvee \left\{ c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}) \,\middle|\, \begin{array}{l} c_1(\bar{x}, \bar{y}) \models C_1(\bar{x}, \bar{y}), \\ c_2(\bar{x}, \bar{y}) \models C_2(\bar{x}, \bar{y}) \end{array} \right\}.$$

The following two lemmata justify viewing sets as disjunctions.

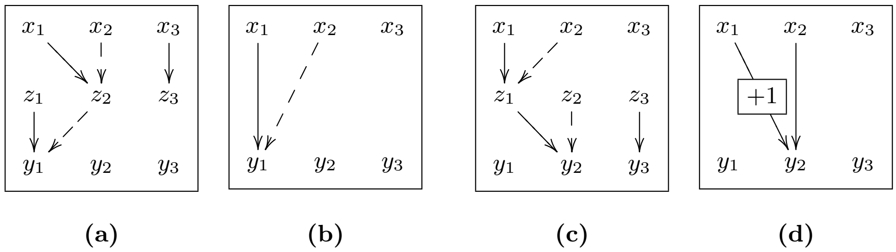


Fig. 4. Constraints of Example 5: **(a-b)** $c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}) = \exists \bar{z}.c_1(\bar{x}, \bar{z}) \wedge c_2(\bar{z}, \bar{y})$; **(c-d)** $c_2(\bar{x}, \bar{y}) \circ c_1(\bar{x}, \bar{y}) \neq \exists \bar{z}.c_2(\bar{x}, \bar{z}) \wedge c_1(\bar{z}, \bar{y})$. (since $\exists z_1.(x_1>z_1) \wedge (z_1>y_2) \equiv x_1+1 > y_2$)

Lemma 1 (disjunctive termination). *Consider a set of size-change graphs $G_{p,p}$ of the form $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), p(\bar{y})$ Then, all graphs in $G_{p,p}$ satisfy the ranking constraint of Definition 6 if and only if $G^{\vee}_{p,p} \models \mathcal{R}(\bar{x}, \bar{y})$.*

Proof. Follows from $(a \vee b) \models c$ if and only if $a \models c$ and $b \models c$.

Lemma 2 (disjunctive composition). *Let $G_{p,q}$ and $G_{q,r}$ be sets of size-change graphs. Then, $(G_{p,q} \circ G_{q,r})^{\vee} \leftrightarrow G^{\vee}_{p,q} \circ G^{\vee}_{q,r}$.*

Proof. Let s be a solution of $(G_{p,q} \circ G_{q,r})^{\vee}$. So there are graphs $g_1 \in G_{p,q}$ and $g_2 \in G_{q,r}$ such that s is a solution of $g_1 \circ g_2$. But g_1 and g_2 respectively entailed $G^{\vee}_{p,q}$ and $G^{\vee}_{q,r}$ and hence s is also a solution for $G^{\vee}_{p,q} \circ G^{\vee}_{q,r}$. The other direction is similar.

From here on we will not distinguish sets from disjunctions. We will view sets of constraints modulo disjunction. We now proceed to provide a more practical way to implement the composition of sets of size-change graphs. The following example provides the intuition and motivation.

Example 5. Figure 4(a-b) illustrates the composition of individual size change graphs $c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}) = c_2(\bar{x}, \bar{y})$ for the size change graphs of Figure 2. The result of the composition is equivalent to the projected conjunction of the original constraints: $\exists \bar{z}.\ c_1(\bar{x}, \bar{z}) \wedge c_2(\bar{z}, \bar{y})$. This correspondence follows because the relations $>$ and $\geq$ satisfy $> \circ \geq\ =\ >$ and $\geq \circ \geq\ =\ \geq$. This gives hope that we might define the composition of size change graphs in terms of renaming, conjunction and projection: $c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y}) = \exists \bar{z}.\ c_1(\bar{x}, \bar{z}) \wedge c_2(\bar{z}, \bar{y})$, and lift the resulting operations to sets defined by disjunctions. However the correspondence does not hold when considering the composition of strict relations such as in $c_2(\bar{x}, \bar{y}) \circ c_1(\bar{x}, \bar{y}) = c_3(\bar{x}, \bar{y})$ as depicted in Figure 4(c-d). The corresponding conjunction is $\exists \bar{z}.\ c_2(\bar{x}, \bar{z}) \wedge c_1(\bar{z}, \bar{y})$ which is equivalent to $x_1 > y_2 + 1 \wedge x_2 > y_2$. The problem is with the constraint $x_1 > y_2 + 1$ which is not of the form $x_i \succ^b y_j$ and the source of the problem is that $> \circ > \neq >$.

We now refine the interpretation of the constraints in a size-change graph so that composition as well as set based composition can be defined in terms of renaming, conjunction and projection. The key is to weaken the greater than relation.

Definition 10 (weak greater-than). *The binary relation $\gg$ over the non-negative integers is given by $\gg\, = \,>\, \cup \{(a,a) \mid a \text{ is even}\}$.*

The intuition behind $\gg$ is the follows: It is stronger than $\geq$ yet weaker than $>$. The projected conjunction $\exists z.(x > z \wedge z > y)$ is not equivalent to $x > y$ because it misses the tuples $(n+1, n)$. But one of $n+1$ or n is even. Hence, using $\gg$ we can assign z to the even value, and we have that $x \gg y \leftrightarrow \exists z.(x \gg z \wedge z \gg y)$.

Definition 11 (size-change graphs - II). *Reconsider Definition 1 of a size-change graph and Definition 6 of the ranking constraint. But this time the relations $\succ^1$ and $\succ^0$ are interpreted as $\gg$ and $\geq$.*

Lemma 3. *Lemma 1 is not influenced when $\succ^1$ is interpreted as $\gg$ instead of as $>$ in Definitions 8 and 6.*

Proof. (Sketch) For an individual graph $c(\bar{x}, \bar{y}) \models \mathcal{R}(\bar{x}, \bar{y})$ is equivalent to showing there are no solutions of $c(\bar{x}, \bar{y}) \wedge \wedge_{i=1}^{n}(\neg x_i \succ^1 y_i)$. For $\succ^1 \equiv >$, this means finding a loop including a strict arc in the size change graph $c(\bar{x}, \bar{y})$ with $\geq$ arcs added from each y_i to x_i (since $\neg x_i > y_i \leftrightarrow y_i \geq x_i$). For $\succ^1 \equiv \gg$ this amounts to the same thing except the upwards arcs are $\overline{\gg} \;\equiv\; > \cup \{(a,a) \mid a \text{ is odd}\}$. Clearly a loop including a $\gg$ arc and $\overline{\gg}$ arc is not satisfiable since values taken by variables in a solution must be nonincreasing, and hence identical around the loop, but then the $\gg$ arc requires that the value is even while $\overline{\gg}$ requires it is odd.

In the remainder of the paper, size-change graphs and termination constraints are to be interpreted in terms of $\gg$ and $\geq$ unless stated otherwise. We are now in position to obtain set-based composition as conjunction.

Lemma 4 (set based composition). *Let $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ be the disjunctive representations of sets of size-change graphs G_1 and G_2 respectively of the forms $p(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), q(\bar{y})$ and $q(\bar{x}) \leftarrow c(\bar{x}, \bar{y}), r(\bar{y})$. Then,*

$$C_1(\bar{x}, \bar{y}) \circ C_2(\bar{x}, \bar{y}) = \exists \bar{z}.(C_1(\bar{x}, \bar{z})) \wedge (C_2(\bar{z}, \bar{y})).$$

Proof. After distributing $\wedge$ over $\vee$ it is left to show that the claim holds for individual disjuncts $c_1(\bar{x}, \bar{y})$ and $c_2(\bar{x}, \bar{y})$. This follows because $\gg \circ \gg \, = \, \gg$ and $\gg \circ \geq \; = \; \geq \circ \gg \; = \; \gg$.

Example 6. Consider the composition of the graphs depicted in Figure 4(c). We have $c_1(\bar{x}, \bar{y}) = (x_1 \gg y_1) \wedge (x_2 \geq y_1)$ and $c_2(\bar{x}, \bar{y}) = (x_1 \gg y_2) \wedge (x_2 \geq y_2) \wedge (x_3 \gg y_3)$. Consider the problematic (renamed) pair of relations $(x_1 \gg z_1)$ from c_1 and $(z_1 \gg y_2)$ from c_2. The projected conjunction $\exists z_1.(x_1 \gg z_1) \wedge (z_1 \gg y_2)$ in the composition $c_1(\bar{x}, \bar{y}) \circ c_2(\bar{x}, \bar{y})$ now results in $x_1 \gg y_2$ as required.

This completes the theoretical specification of all of the components required to perform set-based size change termination analysis. To make this practical we still have to provide an adequate data structure to represent sets of size-change graphs and support the set-based operations.

4 Finite domain size change graphs

We proceed to design an analyzer which computes the closure under composition of the given set of size-change graphs and then tests each disjunctive constraint $C(\bar{x}, \bar{y})$ in the closure for the existence of a ranking function using the test $C(\bar{x}, \bar{y}) \models \mathcal{R}(\bar{x}, \bar{y})$ as provided by Lemma 1.

One idea is to apply a general-purpose constraint solver such as CLP(R) [7]. This is the choice taken in TerminWeb [4]. The problem is that CLP(R) does not handle natively the disjunctions found in size-change constraints. TerminWeb represents the disjunctions of size change graphs as sets of binary clauses, and implements set-based operations by considering individual disjuncts.

The alternative approach presented in this paper is based on modeling (disjunctive) size-change graphs by *finite-domain constraints*. All atomic operations of the analyzer take sets as objects.

To obtain a representation based on finite domain constraints we define the *restriction* of a constraint to a finite non-negative integer domain.

Definition 12 (domain restriction). *The restriction of a (size-change) constraint $C(\bar{x}, \bar{y})$ to the first d non-negative integers $[0 \dots d-1]$ is denoted by $[C(\bar{x}, \bar{y})]_d$ and given by:*

$$[C(\bar{x}, \bar{y})]_d \equiv C(\bar{x}, \bar{y}) \wedge \bigwedge_{i=1 \dots n} x_i, y_i \in [0 \dots d-1]$$

For all practical purposes there is no loss of information when restricting sets of size change graphs to a sufficiently large domain. The intuition is that for any solution of a set of size change graphs with d nodes, the same ordering between the values can be represented with only d different non-negative integers. However, there are subtleties. The next three lemmata illustrate that the representation and operations are preserved.

Lemma 5. *Let $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ be disjunctive size-change constraints with $|\bar{x}| = m$ and $|\bar{y}| = n$. Then $C_1(\bar{x}, \bar{y})$ is equivalent to $C_2(\bar{x}, \bar{y})$ if and only if $[C_1(\bar{x}, \bar{y})]_{m+n}$ is equivalent to $[C_2(\bar{x}, \bar{y})]_{m+n}$.*

Proof. If the constraints are equivalent then clearly the corresponding restrictions are as well. Consider the opposite direction and assume for the purpose of contradiction that $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ define the same sets of solutions over the domain of $m+n$ values, but differ in at least one solution over the infinite non-negative integer domain. Assume without loss of generality that $\theta = \{x_1/v_1, \dots, x_m/v_m,\ y_1/v_{m+1}, \dots, y_n/v_{m+n}\}$ is a solution of $C_1(\bar{x}, \bar{y})$ but not of $C_2(\bar{x}, \bar{y})$. Consider first the case where size change graphs are interpreted in terms of the binary relations $>$ and $\geq$. Define a solution θ' which maps each variable of $\bar{x}$ and $\bar{y}$ to its index in the ascending order induced on the corresponding values $\{v_1, \dots, v_{m+n}\}$. We also make sure that two (or more) variables mapped by θ to the same value v are mapped to the same index by θ'. Clearly, all pairwise relations imposed on $\bar{x}$ and $\bar{y}$ by θ are preserved intact by θ'. Thus,

θ' is a solution of $C_1(\bar{x}, \bar{y})$ but not of $C_2(\bar{x}, \bar{y})$ in contradiction to the assumption that they define the same set of solutions over the domain of $m + n$ values.

Now consider the case where size change graphs are interpreted over the relations $\gg$ and $\geq$. We have an additional requirement on θ' from the previous case. If for a pair of variables $x, y \in \bar{x} \cup \bar{y}$ we have $\theta(x) = \theta(y) = v$ and the corresponding $\theta'(x) = \theta'(y) = v'$ then we require that v' is even if and only if v is even. Note that the domain of $m + n$ distinct values is still sufficient for defining θ'.

Lemma 6 (finite-domain termination test). *Let $p_{/n}(\bar{x}) \leftarrow C(\bar{x}, \bar{y}), p_{/n}(\bar{y})$ be a disjunctive (recursive) size change graph. Then,*

$$C(\bar{x}, \bar{y}) \models \mathcal{R}(\bar{x}, \bar{y}) \quad \Leftrightarrow \quad [C(\bar{x}, \bar{y})]_{2n} \models [\mathcal{R}(\bar{x}, \bar{y})]_{2n}$$

Proof (sketch). We need to show that $C(\bar{x}, \bar{y}) \wedge \neg\mathcal{R}(\bar{x}, \bar{y})$ is satisfiable if and only if $[C(\bar{x}, \bar{y}) \wedge \neg\mathcal{R}(\bar{x}, \bar{y})]_{2n}$ is satisfiable. The constraint $C(\bar{x}, \bar{y}) \wedge \neg\mathcal{R}(\bar{x}, \bar{y}) = C(\bar{x}, \bar{y}) \wedge \bigwedge_{i=1}^{n} \neg(x_i \gg y_i)$ is a constraint based on pairwise order relations between the elements of $\bar{x}$ and $\bar{y}$. We assume a solution θ of that constraint and show using the same mapping as in the proof of Lemma 5 that the constraint is satisfiable if and only if it is satisfiable over the domain of $2n$ elements.[4] The claim follows by observing (through the straightforward transformation) that $[C(\bar{x}, \bar{y})]_{2n} \models [\mathcal{R}(\bar{x}, \bar{y})]_{2n}$ is equivalent to $[C(\bar{x}, \bar{y}) \models \mathcal{R}(\bar{x}, \bar{y})]_{2n}$.

Lemma 7 (finite domain composition). *Let $C_1(\bar{x}, \bar{z})$ and $C_2(\bar{z}, \bar{y})$ be disjunctive size-change constraints with $|\bar{x}| = m$ and $|\bar{y}| = n$. Then*

$$[C_1(\bar{x}, \bar{z})]_{m+n} \circ [C_2(\bar{z}, \bar{y})]_{m+n} = [C_1(\bar{x}, \bar{z}) \circ C_2(\bar{z}, \bar{y})]_{m+n}$$

Proof (sketch). The proof technique is similar to that of Lemma 5. We assume a solution ϕ on $\bar{x} \cup \bar{y}$ of $C_1(\bar{x}, \bar{z}) \circ C_2(\bar{z}, \bar{y})$. By definition there exist $c_1(\bar{x}, \bar{z}) \models C_1(\bar{x}, \bar{z})$ and $c_2(\bar{z}, \bar{y}) \models C_2(\bar{z}, \bar{y})$ and solution

$$\theta = \{x_1/v_{x1}, \ldots, x_m/v_{xm},\ y_1/v_{y1}, \ldots, y_n/v_{yn},\ z_1/v_{z1}, \ldots, z_l/v_{zl}\}$$

of the conjunction $c_1(\bar{x}, \bar{z}) \wedge c_2(\bar{z}, \bar{y})$ extending ϕ (i.e. $\phi(v) = \theta(v), v \in \bar{x} \cup \bar{y}$) and thus, of each of $c_1(\bar{x}, \bar{z})$ and $c_2(\bar{z}, \bar{y})$ individually. We show that this solution can be transformed to another solution θ' with at most $m + n$ distinct values in the range, yet preserving the "$\gg$"-order relations for each pair $(x, z) \in (\bar{x} \times \bar{z})$ and $(z, y) \in (\bar{z} \times \bar{y})$. So θ' is a solution of $[C_1(\bar{x}, \bar{z})]_{m+n}$ and $[C_2(\bar{z}, \bar{y})]_{m+n}$. The transformation starts by ordering the variables in $\bar{x}$, $\bar{y}$ and $\bar{z}$ with respect to their assigned values v_i. Then we "shift" the values for the variables of $\bar{x}$ and $\bar{y}$ until each variable of $\bar{z}$ shares its assigned value with either a variable of $\bar{x}$ or a variable of $\bar{y}$. The transformation is always possible and thus, it is sufficient to prove the claim only for the solutions with at most $m + n$ distinct values in the range of the substitution. In that case the operation of domain restriction $[\cdot]_{m+n}$ degenerates to an identity, and the two parts of the formula in the claim become the same.

[4] Note that $C(\bar{x}, \bar{y}) \wedge \neg\mathcal{R}(\bar{x}, \bar{y})$ is not a size-change graph and thus, it is not always satisfiable. However, the proof technique of Lemma 5 applies to any constraints based on pairwise order relations.

5 Size change termination with k bits

To facilitate efficient size change termination analysis we observe that in practice it is often sufficient to interpret size change constraints over a finite domain with a smaller number of values than nodes in the graphs. As our implementation is based on a Boolean representation of finite domain constraints we will consider values in binary form and typically chose a number of values d of the form $d = 2^k$. Experimental results indicate that all of the size change termination problems in our benchmark suite are guaranteed to be analysed correctly using a 2-bit representation. The following example illustrates the main idea.

Example 7. Consider the disjunctive representation for the graphs in Figure 2:

$$(x_1 \gg y_2 \wedge x_2 \geq y_2 \wedge x_3 \gg y_3) \vee (x_1 \gg y_1 \wedge x_2 \geq y_1) \vee (x_1 \gg y_2 \wedge x_2 \gg y_2).$$

From the results of the previous section we know that for termination analysis we can consider solutions over 6 values. However, note that the constraints in this example are "partitioned" in two blocks of nodes: $I = \{x_1, x_2, y_1, y_2\}$ and $J = \{x_3, y_3\}$. There are no constraints linking the nodes of I and J. As we shall see we can interpret the constraint over the domain of 4 elements i.e., the size of the larger partition.

Definition 13 (partitioning size-change constraints). *We say that a (disjunctive) size-change constraint $C(\bar{x}, \bar{y})$ can be partitioned if the set of arguments $\bar{x} \cup \bar{y}$ can be partitioned into two disjoint non-trivial subsets I and J such that $C(\bar{x}, \bar{y}) \equiv (\exists I.C(\bar{x}, \bar{y})) \wedge (\exists J.C(\bar{x}, \bar{y}))$.*

We proceed to formalize the intuition of Example 7.

Lemma 8. *Let $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ be size-change graphs that admit the same partitioning induced by the sets I and J of nodes. Then $C_1(\bar{x}, \bar{y})$ is equivalent to $C_2(\bar{x}, \bar{y})$ if and only if $[C_1(\bar{x}, \bar{y})]_{\max(|I|,|J|)}$ is equivalent to $[C_2(\bar{x}, \bar{y})]_{\max(|I|,|J|)}$.*

Proof. Assume for the purpose of contradiction that $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ are not equivalent while $[C_1(\bar{x}, \bar{y})]_{\max(|I|,|J|)}$ and $[C_2(\bar{x}, \bar{y})]_{\max(|I|,|J|)}$ are. That means that either $\exists I.C_1(\bar{x}, \bar{y})$ is not equivalent to $\exists I.C_2(\bar{x}, \bar{y})$ or $\exists J.C_1(\bar{x}, \bar{y})$ is not equivalent to $\exists J.C_2(\bar{x}, \bar{y})$. By Lemma 5 the equivalence of $\exists I.C_1(\bar{x}, \bar{y})$ and $\exists I.C_2(\bar{x}, \bar{y})$ can be tested using a finite domain of at most $|J|$ elements. Similarly, the equivalence of $\exists J.C_1(\bar{x}, \bar{y})$ and $\exists J.C_2(\bar{x}, \bar{y})$ can be tested using a finite domain of at most $|I|$ elements. Thus, if $C_1(\bar{x}, \bar{y})$ and $C_2(\bar{x}, \bar{y})$ are not equivalent, then there must be a substitution over the domain of (at most) $\max(|I|, |J|)$ elements which distinguishes between the two constraints. Hence, we have a contradiction.

In a similar way we can tighten the bounds of Lemma 6 and Lemma 7 which show distributivity of domain restriction and the operations (composition and testing for termination). Moreover, for correctness of the analysis there is now an additional operation to consider: partitioning.

Lemma 9. *Let size-change constraint $C(\bar{x},\bar{y})$ admit a partition (I,J) of arguments such that* $\max(|I|,|J|) = m > 2$*. Then the size-change graph $[C(\bar{x},\bar{y})]_m$ admits the same partition.*

Lemma 9 does not hold for $m = 2$ because a constraint $x \gg y$ for $x, y \in I$ in the two-value domain implies $y = 0$ and hence, $x' \gg y$ for any x', including $x' \in J$ (and vice versa). Hence, the restriction to two values introduces new dependencies between the elements of I and J. For $m > 2$ and without loss of generality for $x \in I$ and $y \in J$, we can assign values to x and y so that either $x \gg y$ or $\neg(x \gg y)$ holds.

In a recent paper [1] Ben-Amram and Lee provide the following definitions which we will make use of.

Definition 14 (size relation graph [1]). *Let G be a set of size change graphs. The corresponding size-relation graph, denoted $srg(G)$, is the annotated digraph with vertex set $Par(G)$ and an edge from $p_{\langle i\rangle}$ to $q_{\langle j\rangle}$ labelled by $\langle b, g\rangle$ if $g = \; p_{/n}(\bar{x}) \leftarrow c(\bar{x},\bar{y}), q_{/m}(\bar{y})$ is a graph in G and $x_i \succ^b y_j$ an edge in g.*

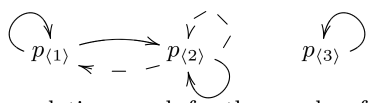


Fig. 5. Size relation graph for the graphs of Figure 2.

Definition 15 ($clean(G)$ [1]). *For a set G of size change graphs, $clean(G)$ is the set of graphs G minus every arc not belonging to a strongly connected component of $srg(G)$ that contains a label $b = 1$. If $G = clean(G)$ we say that G is clean.*

Example 8. Consider the set of graphs G from Figure 2. The corresponding size-relation graph $srg(G)$ is depicted as Figure 5. Observe that G is clean.

The following lemma enables us to restrict attention to sets of cleaned size-change graphs.

Lemma 10 ([1]). *A set of size change graphs G satisfies size change termination if and only if $clean(G)$ does.*

To determine the number of bits required to perform size change termination analysis for a set of graphs it is sufficient to check the size of the largest strongest connected component in $srg(clean(G))$.

Definition 16 (diameter). *Let G be a set of size change graphs. The diameter of G is the largest number of parameters with the same predicate symbol in a strongly connected component of $srg(clean(G))$.*

Example 9. The diameter of the set of graphs depicted in Figure 2 is 2.

For a set of size-change graphs G, the strongly connected components of $srg(clean(G))$ indicate a partitioning of the nodes of G. This provides a safe bound on the number of values required to represent G.

6 Implementation and experimentation

Boolean encoding: We first illustrate how the binary relations $\geq$ and $\gg$ are modelled for k-bit non-negative integers. Let $\langle v_{k-1}, \ldots, v_0 \rangle$ denote the k-bit binary representation of non-negative integer variable v with left most significant binary digit. The k-bit relation $v \geq w$ is standardly modelled inductively by the following Boolean function:

$$
\begin{array}{l}
\langle\rangle \geq \langle\rangle \equiv 1 \ (true) \\
\langle v_{k-1}, \ldots, v_0 \rangle \geq \langle w_{k-1}, \ldots, w_0 \rangle \equiv (v_{k-1} \wedge \neg w_{k-1}) \bigvee \\
\qquad ((v_{k-1} \leftrightarrow w_{k-1}) \wedge \langle v_{k-2}, \ldots, v_0 \rangle \geq \langle w_{k-2}, \ldots, w_0 \rangle)
\end{array}
$$

The non-standard relation $v \gg w$ of Definition 10 is modelled as

$$v \gg w \equiv v \geq w \wedge ((v \neq w) \vee even(v))$$

where $(v \neq w) \equiv \neg \bigwedge_{i=1}^{n} (v_i \leftrightarrow w_i)$ and $even(v) \equiv \neg v_0$ (the least significant bit is 0). Note that the above formula is equivalent to Definition 10.

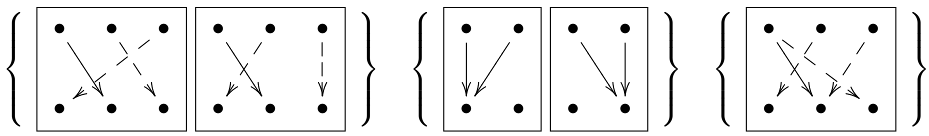
Example 10. For $k = 2$ the relation $x \geq y$ is modelled by $(x_1 \wedge \neg y_1) \vee ((x_1 \leftrightarrow y_1) \wedge (x_0 \rightarrow y_0))$. Note that the models of this formula correspond to the solutions of the constraint $x \geq y$ on the set of four values $\{0, \ldots, 3\}$.

The Boolean encodings of sets of size-change graphs and their set-based operations are obtained from the encoding of the binary relations given above and the set-based definitions of Sections 2 and 3. Size-change graphs are modelled as conjunctions of binary relations. Sets of size-change graphs are modelled as disjunctions of the models of individual size-change graphs. Composition of sets of size-change graphs is modelled through renaming, conjunction and projection. Finally, testing a set of size-change graphs for termination amounts to checking the entailment on two Boolean formula.

A key strength of our approach is that all of the components of the size-change termination analysis can be represented as Boolean formula and standard Boolean operations. This facilitates an implementation based on well-studied data structures and well-engineered tools for representing and manipulating Boolean formulæ.

Prototype Implementation: To validate our ideas, we have constructed a prototype size-change termination analyser. We use *Reduced Ordered Binary Decision Diagrams* [2] (ROBDDs, often just called BDDs) to represent size-change graphs. ROBDDs are a standard — perhaps *the* standard — representation of Boolean formulae for many applications. They have been applied successfully to representation of sets of constraints over a finite domain [6]. In the context of this work we apply ROBDDs to represent sets of size-change graphs and the respective set-based operations.

Our analyzer comprises about 500 lines of Prolog code and is implemented in SWI-Prolog [10]. It utilizes the freely available CUDD [5] package as a back-end for manipulating BDDs and a previously developed module interfacing CUDD with SWI-Prolog (around 650 lines of C-code.)



(a) (b) (c)

Fig. 6. one-to-one, fan-in and fan-out

Results: The analyzer has been applied to a benchmark suite consisting of 339 size-change termination problems. These problems have been generated from the benchmark suite of TerminWeb [9]. The problems can be obtained from `http://www.cs.bgu.ac.il/~mcodish/TerminWeb/scg.tgz`. All 339 problems have diameter two or less. The total analysis time for the benchmarking suite and a 2-bit representation is 1.2 CPU seconds on a 1GHz machine running GNU/Linux 2.4. The longest running single test takes 70 milliseconds. Preliminary comparison indicates that the performance of our analyzer is far superior to the corresponding components of TerminWeb (orders of magnitude).

We note that our analyzer can also handle hard instances of the underlying PSPACE-complete problem. For instance, the example used in the proof of PSPACE-hardness in Theorem 5 of [8] takes 0.4 second to analyze. Unlike the benchmarks of TerminWeb this example has a diameter of 5 and thus, 3-bit encodings of graph nodes are required for its analysis.

7 Related work

Ben-Amram and Lee introduce a polynomial algorithm (termed SCP) which covers many instances of size-change termination [1]. SCP is shown to be complete for sets of size-change graphs which are "one-to-one" (the in- and out-degree of all nodes is not more than 1). Their basic SCP algorithm is correct but not complete for sets of graphs which are "fan-in free" (the in-degree of all nodes is not more than 1) and several techniques to handle certain kinds of graphs with "fan-in" are also proposed. Experimental evaluation indicates that SCP is complete for their benchmark suite (circa 90 SCT problems).

The set of graphs depicted as Figure 6(a) is one-to-one. The SCP algorithm will detect that these graphs satisfy size change termination in polynomial time without computing the expensive closure operation. The graphs in Figure 6(b) have fan-in and SCP does not detect that they are terminating. The graph in Figure 6(c) has both fan-in and fan-out. Its termination also cannot be detected using SCP.

In contrast, our k-bit representation is always complete for any set of size change graphs with diameter 2^{k-1} or less. Experimental evaluation indicates that for our benchmark suite (which extends the one used by Ben-Amram and Lee and consists of 339 SCT problems) all of the examples have diameter 2 or less (after cleanup). Hence the 2-bit size-change termination analysis is guaranteed to be complete. For the examples of Figure 6 our technique requires a 3-bit analysis for (a) and (c) and a 2-bit analysis for (b).

8 Conclusion

This paper proposes a constraint-based approach to size-change termination analysis. We model size-change graphs, sets of size-change graphs and operations for size-change termination using Boolean functions. We draw on experience from Boolean functions where representing large sets of models is well studied. A key step in our design is the non-standard interpretation of size-change relations "$>$" and "$\geq$". This enables us to encode union and composition of sets of size-change graphs by disjunction and conjunction. The proposed approach has been implemented using BDD-based modeling and BDD operations. The initial performance indicators are highly encouraging.

Acknowledgement

We thank Samir Genaim for generating the size-change termination problems from the benchmark suite of TerminWeb, and the referees for their insightful comments.

References

1. A. M. Ben-Amram and C. S. Lee. Size-change termination in polynomial time. submitted for publication, 2004.
2. R. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.
3. M. Codish, V. Lagoon, and P. J. Stuckey. Testing for termination with monotonicity constraints. In G. Gabbrielli, Maurizio; Gupta, editor, *21st International Conference, ICLP*, volume 3668 of *LNCS*. Springer, October 2005.
4. M. Codish and C. Taboch. A semantic basis for the termination analysis of logic programs. *The Journal of Logic Programming*, 41(1):103–123, 1999.
5. http://vlsi.colorado.edu/˜fabio/CUDD/.
6. P. Hawkins, V. Lagoon, and P. J. Stuckey. Solving set constraint satisfaction problems using ROBDD's. *Journal of Artificial Intelligence Research*, 24:106–156, 2005.
7. J. Jaffar, S. Michaylov, P. J. Stuckey, and R. H. C. Yap. The CLP($\mathcal{R}$) language and system. *ACM TOPLAS*, 14(3):339–395, 1992.
8. C. S. Lee, N. D. Jones, and A. M. Ben-Amram. The size-change principle for program termination. *ACM SIGPLAN Notices*, 36(3):81–92, 2001. Proceedings of POPL'01.
9. C. Taboch, S. Genaim, and M. Codish. Terminweb: Semantic based termination analyser for logic programs, 1999-2002. `http://www.cs.bgu.ac.il/~mcodish/TerminWeb`.
10. J. Wielemaker. An overview of the SWI-Prolog programming environment. In F. Mesnard and A. Serebenik, editors, *Proceedings of the 13th International Workshop on Logic Programming Environments*, pages 1–16, Heverlee, Belgium, Dec. 2003. Katholieke Universiteit Leuven. CW 371.