A Framework for Combining Set Variable Representations

Christian Bessiere  
CNRS  
University of Montpellier, France  
bessiere@lirmm.fr

Zeynep Kiziltan, Andrea Rappini  
DISI  
University of Bologna, Italy  
{zeynep.rappini}@cs.unibo.it

Toby Walsh  
NICTA and University of New South Wales  
Australia  
toby.walsh@nicta.com.au

Abstract

Set and multiset variables are important modelling constructs in constraint programming. Several representations have been proposed for set and multiset variables, often based on combining together different representations. In this paper, we provide a formal framework with which we can study many existing combinations of representations and compare their strength. In addition, our framework opens the door to interesting new combinations, as well as to the construction of propagators with well defined properties. We illustrate the value of the framework via both theoretical and experimental results.

1 Introduction

Many constraint satisfaction problems can be modelled naturally using set variables. Consider for instance the social golfers problem (prob010 at CSPLib.org) which partitions a set of golfers into foursomes. Another example is the ternary Steiner problem from combinatorial mathematics which requires to find triples of integers from 1 to \( n \) (Lindner and Rosa 1980). Several constraint toolkits therefore provide set variables as modelling and solving construct since the earliest constraint solvers (Puget 1992; Gervet 1994).

An important issue regarding a set variable is how to represent it. As an extensional representation of the domain of a set variable is exponential, more compact representations are typically used which approximate the full powerset domain. The first representation proposed for set variables, which is still used in many solvers, is subset bounds. The lower bound is the set of definite elements, whilst the upper bound is the set of possible elements. Unfortunately this is a weak approximation. To better approximate domains, new representations have been proposed that combine together different representations (e.g. (Müller 2001; Azevedo 2007; Gervet and van Hentenryck 2006; Sadler and Gervet 2008; Malitsky, Sellmann, and van Hoeve 2008)). For instance, set constraint programming in Oz (Müller 2001) and the Cardinal set solver (Azevedo 2007) both reason with subset and cardinality bounds on the set. Similar issues arise with multiset variables. In (Law, Lee, and Woo 2009), for instance, cardinality and variety bounds are combined with subset bounds (Kiziltan and Walsh 2002).

In this paper, we present a formal framework with which we can study and compare such combined representations of sets and multisets. We define combination operators, notions of consistency between representations, and of propagation. Using this framework, we launch a systematic study of representations. Due to lack of space, we focus only on set variables but our results are directly applicable to multisets. Our study provides insight into existing combined representations, as well as new and potentially useful representations. Moreover, it uncovers a number of important relations. For example, we prove that the lengthlex representation for set variables (which combines lexicographic ordering and cardinality information and is designed to work well with cardinality and ordering constraints) is incomparable to a representation for set variables that deals with the lexicographic ordering and cardinality bounds either separately or simultaneously. As a second example, we prove that the hybrid representation of (Sadler and Gervet 2008) (which combines lexicographic, subset and cardinality bounds) is tighter than one that looks at pairs of bounds separately, but looser than one that looks at all the bounds simultaneously.

We demonstrate the value of our framework via theoretical and experimental results. We show that, whilst it can be intractable in general to combine representations, some combinations are powerful but polynomial to reason about. For example, for some common set constraints, reasoning with subset and cardinality bounds simultaneously is not more difficult than reasoning about them separately (as Oz and Cardinal do). Our experiments demonstrate that combining another representation with subset bounds can result in a tight approximation. Hence, one of our conclusions is that developers of solvers should consider more tightly integrating subset bounds with representations like cardinality and lengthlex.

2 Background

A set variable takes as value a subset of a universe \( U = \{1, \ldots, n\} \). The domain may contain up to \( 2^n \) values. The representation \( R \) of a set domain is typically given by bounds that limit the domain to an interval in a (partial or total) order of the subsets of \( 2^U \). The approximation \( D_R(S) \)
of a domain is given by a lower and upper bound, \( lb_D(S) \) and \( ub_D(S) \).

The **subsetbounds** representation (denoted \( SB \)) uses the partial order over subsets (Puget 1992; Gervet 1994). The lower bound contains the definite elements, and the upper bound contains the possible elements. Given a variable \( S \), \( D_{SB}(S) = \{ S \mid lb_{SB}(S) \subseteq S \subseteq ub_{SB}(S) \} \). Given a set of sets \( A \subseteq 2^U \), we define \( glb_{SB}(A) = \bigcap_{a \in A} a \) and \( lub_{SB}(A) = \bigcup_{a \in A} a \).

Another approach is to consider lower and upper bounds, \( lb_T \) and \( ub_T \) within a total ordering \( \preceq_T \) on \( 2^U \). As before, \( D_T(S) = \{ S \mid lb_T(S) \preceq_T S \preceq_T ub_T(S) \} \). In **minlex** (denoted \( ML \)), the total ordering \( \preceq_{ML} \) is defined as follows: we list elements in each set from smallest to largest and then order lists obtained in this way lexicographically. In **lengthlex** (denoted \( LL \) (Gervet and van Hentenryck 2006)), the total ordering \( \preceq_{LL} \) first orders sets by size, breaking ties with \( \preceq_{ML} \). In the hybrid domain of (Sadler and Gervet 2008), the total ordering \( \preceq_{maxlex} \) is defined as follows: we list elements in each set from largest to smallest and then order lists obtained in this way lexicographically. Given a set of sets \( A \subseteq 2^U \), we define \( glb_T(A) = \min_{\preceq_T}(A) \) and \( lub_T(A) = \max_{\preceq_T}(A) \).

A third approach is based on the size or value of a set. We illustrate this with the cardinality representation (card) but we could also consider cost of the set. In card, \( D_C(S) = \{ S \mid lb_C(S) \leq |S| \leq ub_C(S) \} \). Given a set of sets \( A \subseteq 2^U \), we define \( glb_C(A) = \min\{|a| \mid a \in A\} \) and \( lub_C(A) = \max\{|a| \mid a \in A\} \).

**Example 1** Consider a variable \( S \) whose domain \( D(S) = \{1, 1.2, 3\} \). Given a representation \( R \) among subsetbounds, **minlex**, **lengthlex** and card, the domain \( D_R(S) \) is defined as follows:
- \( D_{SB}(R) = \{1, 2, 3\} \) and \( ub_{SB}(R) = \{1, 2, 3\} \);
- \( D_{ML}(S) = \{3\} \) and \( ub_{ML}(S) = \{3\} \);
- \( D_{LL}(S) = \{1, 2\} \) and \( ub_{LL}(S) = \{1, 2\} \);
- \( D_C(S) = \{1\} \) and \( ub_C(S) = \{2\} \).

We now give an uniform definition of bound consistency (BC). Given representation \( R \), variables \( S_1, \ldots, S_n \) and a constraint \( c(S_1, \ldots, S_n) \), we denote by \( c(S_1, \ldots, S_n) \) the set \( \{ s_i \mid c(s_1, \ldots, s_i, \ldots, s_n) \wedge s_j \in D_R(S_j) \forall j \in 1..n \} \).

**Definition 1** (Bound consistency on \( R \)) Given a representation \( R \), variables \( S_1, S_2, \ldots, S_n \) and a constraint \( c(S_1, \ldots, S_n) \), we say that \( S_1 \) is bound consistent on \( c \) for \( R \) iff \( lb_R(S_1) = glb_R(c(S_1|R)) \) and \( ub_R(S_1) = lub_R(c(S_1|R)) \).

**Definition 2** (Synchronization) Given a variable \( S \), and two representations \( R_1 \) and \( R_2 \), then \( R_1 \) is synchronized with the representation \( R_2 \) on \( S \) iff:
- \( lb_{R_1}(S) = glb_{R_2}(D_{R_1}R_2(S)) \);
- \( ub_{R_1}(S) = lub_{R_2}(D_{R_1}R_2(S)) \);

\( S \) is synchronized for \( R_1 \) and \( R_2 \) if \( R_1 \) is synchronized with \( R_2 \) and \( R_2 \) is synchronized with \( R_1 \).

**Example 3** Consider a variable \( S \) with \( U = \{1, 2, 3\} \) represented using \( R_1 \) = **minlex** with \( lb_{ML}(S) = \{1, 2, 3\} \) and \( ub_{ML}(S) = \{3\} \) and using \( R_2 = \) **SB** with \( lb_{SB}(S) = \{1\} \) and \( ub_{SB}(S) = \{3\} \). We have \( D_{ML}(S) = \{1, 2, 3\} \) and \( D_{SB}(S) = \{1, 3\} \). The intersecting domain \( D_{ML}MB(S) = \{1, 3\} \). For **minlex** to be synchronized with **subsetbounds**, its lower bound \( lb_{ML}(S) \) must be \( lb_{ML}(D_{ML}MB(S)) = \{1, 3\} \). Similarly, for **subsetbounds to be synchronized with **minlex**, its lower bound \( lb_{SB}(S) \) must be \( lb_{SB}(D_{ML}MB(S)) = \{3\} \).

We consider two different ways to propagate a combined representation. In the weak sense, we apply bound consistency on each representation independently and synchronize them. In the strong sense, we ensure the lower and upper bounds of a representation are bound consistent on their synchronized domains. Given two representations \( R_1 \) and \( R_2 \), the weak combination will be denoted by \( R_1 \times R_2 \) and the strong combination by \( R_1 \times R_2 \).

**Definition 3** (Bound consistency on \( R_1 \times R_2 \)) Given variables \( S_1, \ldots, S_n \), and a constraint \( c(S_1, \ldots, S_n) \), \( S_i \) is bound consistent on \( c \) for \( R_1 \times R_2 \) iff \( S_i \) is synchronized for \( R_1 \) and \( R_2 \), \( S_i \in \) **BC** on \( c \) for \( R_1 \) and \( S_i \in \) **BC** on \( c \) for \( R_2 \).

**Example 4** Consider a variable \( S \) with \( U = \{1, 2, 3\} \) represented using \( R_1 \) = **minlex** with \( lb_{ML}(S) = \{1, 2, 3\} \) and \( ub_{ML}(S) = \{3\} \), and using \( R_2 = \) **subsetbounds** with \( lb_{SB}(S) = \{1\} \) and \( ub_{SB}(S) = \{1, 2, 3\} \), and the constraint \( c = \{ |S| \} = 2 \). The variable \( S \) is **BC** on \( c \) for **SB** but is not for **minlex**. Enforcing **BC** on \( c \) for **minlex** would update the bounds of **minlex** so that their cardinality is 2, hence we get \( lb_{ML}(S) = \{1, 2\} \) and \( ub_{ML}(S) = \{2, 3\} \).

With these domains, \( S \) is not synchronized for **minlex** and **SB**, because **SB** requires that \( l \) belongs to \( S \). Ensuring synchronization, \( ub_{ML}(S) \) is further updated to \( \{1, 3\} \). Now, \( S \) is **BC** on \( c \) for **minlex** + **SB**.
### Definition 4 (Bound consistency on \( R_1 \times R_2 \))

Given variables \( S_1, \ldots, S_n \) and a constraint \( c(S_1, \ldots, S_n) \), \( S_i \) is bound consistent on \( c \) from \( R_1 \) to \( R_2 \) iff:

- \( lb_{R_1}(S) = glb_{R_1}(c(S_1|R_1,R_2)) \)
- \( ub_{R_1}(S) = lub_{R_1}(c(S_1|R_1,R_2)) \)

\( S_i \) is bound consistent on \( c \) for \( R_1 \times R_2 \) iff \( S_i \) is synchronized for \( R_1 \) and \( R_2 \), and \( S_i \) is bound consistent from \( R_1 \) to \( R_2 \) and vice versa.

### Example 5

Consider two variables \( S_1 \) and \( S_2 \) with \( U = \{1, 2, 3\} \) represented using \( R_1 = \text{minlex} \) with \( lb_{ML}(S_1) = \{1\} \) and \( ub_{ML}(S_1) = \{1\} \), and \( lb_{ML}(S_2) = \{1\} \), \( ub_{ML}(S_2) = \{3\} \). Therefore, \( S_1 \) is bound consistent on \( c \) for \( R_1 \) and \( R_2 \), and vice versa.

4. Comparing Combinations

Informally, one representation is stronger than another if bound consistency reasoning with the first representation is more powerful than with the second.

### Definition 5 (Stronger relation \( \succeq \))

Given two representations \( R_1 \) and \( R_2 \) and a constraint \( c(S_1, \ldots, S_n) \) such that \( \forall i, S_i \) is synchronized for \( R_1 \) and \( R_2 \), we say that \( R_1 \) is stronger than \( R_2 \) on \( c \) (\( R_1 \succeq c R_2 \)) iff \( \text{BC fails on } c \text{ for } R_1 \text{ then BC fails on } c \text{ for } R_2 \).

We say that \( R_1 \) is strictly stronger than \( R_2 \) on \( c \) (\( R_1 \succ c R_2 \)) iff \( R_1 \succeq c R_2 \) and \( R_2 \not\succeq c R_1 \).

We say that \( R_1 \) and \( R_2 \) are equivalent on \( c \) (\( R_1 \equiv c R_2 \)) iff \( R_1 \succeq c R_2 \) and \( R_2 \succeq c R_1 \).

We say that \( R_1 \) and \( R_2 \) are incomparable on \( c \) (\( R_1 \sim c R_2 \)) iff \( R_1 \succeq c R_2 \) and \( R_2 \succeq c R_1 \).

Definitions for comparing propagation are often based on how they prune values (e.g. (Debruyne and Bessiere 1997)). However, such definitions are limited as both representations must represent the same domains (i.e., the representations are isomorphic). By using a definition based on detection of failure, we can compare different representations of domains that are not necessarily isomorphic. This choice is similar to that made in (Bacchus et al. 2002).

The following theorems are useful to compare two (combined) representations.

### Theorem 1

Given two representations \( R_1 \) and \( R_2 \):

- \( (R_1 + R_2) \succeq R_1 \) and \( (R_1 + R_2) \succeq R_2 \)
- \( (R_1 \times R_2) \succeq R_1 \) and \( (R_1 \times R_2) \succeq R_2 \).

**Proof:** Immediate from the definitions. □

### Theorem 2

Given two representations \( R_1 \) and \( R_2 \):

\( (R_1 \times R_2) \succeq (R_1 + R_2) \).

### Proof:

Let \( c(S_1, \ldots, S_n) \) be a constraint. Suppose BC does not fail for \( R_1 \times R_2 \). Definition 4 tells us that BC on \( R_1 \times R_2 \) requires that for every \( S_i \), \( lb_{R_1}(c(S_1|R_1,R_2)) \), \( ub_{R_1}(c(S_1|R_1,R_2)) \), \( lb_{R_2}(c(S_1|R_1,R_2)) \), \( ub_{R_2}(c(S_1|R_1,R_2)) \), and \( lb_{R_1}(c(S_1|R_1,R_2)) \) are synchronized because \( c(S_1|R_1,R_2) \subseteq D(S_1|R_1,R_2) \). These bounds are also obviously BC for \( R_1 \) and BC for \( R_2 \), so BC for \( R_1 + R_2 \). Therefore, if BC does not fail for \( R_1 \times R_2 \), then \( R_1 + R_2 \) cannot fail, as in the worst case, it will converge on the same bounds as for \( R_1 \times R_2 \). □

When representations can represent the same domains, a stronger relation based on the values pruned can be defined. Let \( \succeq^c \) denote the stronger relation in terms of failure as in Definition 5, and \( \equiv^p \) denotes in terms of pruning. We have:

### Theorem 3

Given two representations \( R_1 \) and \( R_2 \), a constraint \( c \):

- \( R_1 \succ^c R_2 \implies R_1 \succ^p R_2 \)
- \( R_1 \equiv^c R_2 \implies R_1 \equiv^p R_2 \)

Pruning based definition would be useful when comparing for instance \( R_1 \times R_2 \) and \( R_1 + R_2 \) where the \( R_1 \) and \( R_2 \) are isomorphic. As an example, consider a unary constraint \( c \).

We have \( \equiv^p_c (R_1 + R_2) \). A counter example where neither representation is defined by a total ordering is given in Section 5. Pruning based definition can also give us more information when \( \equiv^c \) (\( R_1 + R_2 \)). As an example, consider the binary not equals constraint \( e \equiv S_1 \neq S_2 \). We have \( \equiv^p_c (SB \times card) \) but \( SB \times card >^p_c SB + card \), as we will show later in Section 8.

In the rest of the paper, for the specific representations that we consider, we either have \( R_1 \succ^c R_2 \) or \( R_1 \sim^c R_2 \) for all constraints \( c \) (except Section 8 where we compare representations based on a specific constraint). We therefore consider only the failure based definition and stick to the notation \( \succeq \).

### 5 An example

To illustrate the potential of this framework, we consider pairwise combinations of three of the most popular representations for sets: \( SB \), \( card \) and \( \text{minlex} \).

**Synchronization** The synchronization property of Definition 2 gives rise to the following synchronization rules.

### Representations minlex and card

- \( lb_{\text{card}}(S) \leq |lb_{\text{ML}}(S)| \leq ub_{\text{card}}(c(S)) \leq |ub_{\text{ML}}(S)| \leq ub_{\text{card}}(S) \);
- \( lb_{\text{card}}(c(S)) \in \{|s| \mid s \in D_{\text{ML}}(S)|\} \) and \( ub_{\text{card}}(S) \in \{|s| \mid s \in D_{\text{ML}}(S)|\} \).
Representations minlex and subsetbounds

- \( l_{SB}(S) \subseteq l_{ML}(S) \subseteq u_{SB}(S) \) and \( l_{SB}(S) \subseteq u_{ML}(S) \subseteq u_{SB}(S) \);
- \( l_{SB}(S) \supseteq \bigcap_{s \in D_{ML}(S)} \{s\} \) and \( u_{SB}(S) \subseteq \bigcup_{s \in D_{ML}(S)} \{s\} \).

These synchronization rules are polynomial to enforce.

Representations subsetbounds and card

- \( l_{C}(S) = |u_{SB}(S)| \rightarrow l_{SB}(S) = u_{SB}(S) \);
- \( u_{C}(S) = |u_{SB}(S)| \rightarrow u_{SB}(S) = l_{SB}(S) \);
- \( l_{C}(S) \geq |l_{SB}(S)|, u_{C}(S) \leq |u_{SB}(S)| \).

These synchronization rules are polynomial to enforce.

Pairwise Comparison

Theorem 4 (minlex, SB and card pairwise comparisons)
Figure 1 gives the relations that hold when comparing minlex, SB and card pairwise. Similar results hold for maxlex instead of minlex.

![Figure 1: Pairwise comparison of representations minlex, SB and card.](image)

Proof: By Theorem 1 and Theorem 2, we have all the \( \geq \) relations claimed in Theorem 4. We thus only have to show strictness. We highlight the most important results.

To show \((\text{minlex} \times \text{card}) \succ (\text{minlex} + \text{card})\), take the constraint \(S_1 \cup S_2 \supseteq \{1, 3\}\) and \(U = \{1, 2, 3\}\). Suppose \(D_{ML}(S_1) = D_{ML}(S_2) = \{\{1\}, \{1, 2\}, \{1, 2, 3\}, \{1, 3\}, \{2\}\}\). Then \(l_{C}(S_1) = u_{C}(S_1) = l_{C}(S_2) = u_{C}(S_2) = 1\). \(S_1\) and \(S_2\) are synchronized for both representations. BC on \(\text{minlex} \times \text{card}\) fails whilst BC on \(\text{maxlex} + \text{card}\) does not prune.

To show \((\text{minlex} \times \text{SB}) \succ (\text{minlex} + \text{SB})\), take the constraint \(|S_1| + |S_2| = 5\) and \(U = \{1, 2, 3, 4, 5\}\). Suppose \(D_{ML}(S_1) = D_{ML}(S_2) = \{\{1, 5\}, \{2\}, \{2, 3, 4\}, \{2, 3, 4, 5\}, \{2, 3, 5\}, \{2, 4\}, \{2, 4, 5\}, \{2, 5\}, \{3\}\}\). Then \(l_{C}(S_1) = l_{C}(S_2) = 1\). \(u_{SB}(S_1) = u_{SB}(S_2) = \{1, 3, 5\}\). \(S_1\) and \(S_2\) are synchronized for both representations. BC on \(\text{minlex} \times \text{SB}\) fails whilst BC on \(\text{maxlex} + \text{SB}\) does not prune.

To show \((\text{SB} \times \text{card}) \succ (\text{SB} + \text{card})\), we consider a unary constraint. Take the constraint on \(S\) which is satisfied when the sum of its elements is at least 4 and \(U = \{1, 2, 3, 4, 5\}\). Suppose \(l_{SB}(S) = \{\}\) and \(u_{SB}(S) = \{1, 2, 3\}\). Then \(l_{C}(S) = u_{C}(S) = 1\). \(S\) is synchronized for both representations. BC on \(\text{SB} \times \text{card}\) fails whilst BC on \(\text{SB} + \text{card}\) does not prune.

6 Generalizing the Framework

This framework can be generalized to combinations of more than two representations. We can then reason with, for instance, the hybrid domain representation (Sadler and Gervet 2008) and the multi-set representations of (Law, Lee, and Woo 2009; Law et al. 2011). The extension of the strong combination operator is straightforward as we can take the domain of \(S\) in \(R_1 \times R_2\) to be \(D_{R_1 R_2}(S)\). The extension of the weak combination operator is more problematic as the domain of \(S\) in \(R_1 + R_2\) is not defined explicitly. This necessitates a generalization of the definitions of synchronization and BC. In this generalization, we first define when \(S\) is synchronized for \(R_1 \times \ldots \times R_n\) and for \((R_1 \times \ldots \times R_1) + \ldots + (R_k \times \ldots \times R_n)\). Then, we define when \(S\) is BC on \(e\) for these combinations. Finally, we give some algebraic identities that are useful in comparing (combined) representations.

Theorem 5 \((R_1 \times \ldots \times R_k \times R_{k+1} \times \ldots \times R_n) \succ (R_1 \times \ldots \times R_k) + (R_{k+1} \times \ldots \times R_n)\)

Theorem 6 \((R_1 + \ldots + R_n) \succ R_k\) for all \(k \in \{1, \ldots, n\}\).

Theorem 7 \((R_1 \times \ldots \times R_n) \succ R_k\) for all \(k \in \{1, \ldots, n\}\).

For reasons of space, we skip formal definitions and proofs.

To illustrate this generalization, we consider \(\text{minlex, SB, and card}\). By applying the weak and strong combination operators in all possible ways, we obtain 5 different combined representations: \(\text{minlex} \times \text{SB} \times \text{card}, (\text{minlex} \times \text{SB}) + \text{card}, (\text{minlex} \times \text{card}) + \text{SB}, (\text{SB} \times \text{card}) + \text{minlex},\) and \(\text{minlex} + \text{SB} + \text{card}\). The necessary synchronization and BC rules can be obtained by instantiating their formal definitions. In addition, using Theorems 5–7, we can compare the strength of these combinations and contrast with (combined) representations involving a subset of the three representations. In Figure 2, we show how the triple combinations compare. Due to space, we prove here only the results.

Theorem 8 \((\text{minlex} \times \text{SB}) + \text{card} \sim (\text{minlex} \times \text{card}) + \text{SB}\)

Proof: We first show \((\text{minlex} \times \text{card}) + \text{SB} \not\succ (\text{minlex} \times \text{SB}) + \text{card}\). Take the constraint \(|S_1 \cap S_2| = 1\) on variables \(S_1\) and \(S_2\), and \(U = \{1, 2, 3, 4, 5\}\). Suppose \(D_{ML}(S_1) = D_{ML}(S_2) = \{\{1, 5\}, \{2\}, \{2, 3\}, \{2, 3, 4\}, \{2, 3, 4, 5\}, \{2, 3, 5\}, \{2, 4\}, \{2, 4, 5\}, \{2, 5\}, \{3\}, \{3, 4\}\}\). Then \(l_{SB}(S_1) = l_{SB}(S_2) = \{\}\), \(u_{SB}(S_1) = u_{SB}(S_2) = \{1, 3, 5\}\). Then \(l_{C}(S_1) = l_{C}(S_2) = u_{C}(S_1) = u_{C}(S_2) = 2\). \(S_1\) and \(S_2\)
are synchronized for all representations. \((\text{minlex} \times SB) + \text{card}\) fails (\{3\} is pruned thanks to synchronization with \(\text{card}\)) but \((\text{minlex} \times \text{card}) + \text{SB}\) does not prune.

We now show \((\text{minlex} \times SB) + \text{card} \not\subseteq (\text{minlex} \times \text{card}) + SB\). Take the constraint \(S_1 \cup S_2 \supseteq \{2, 3, 4\}\) on variables \(S_1\) and \(S_2\). Suppose \(D_{ML}(S_1) = D_{ML}(S_2) = \{(1, 2), \{1, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4\}\), \(\text{lb}_{SB}(S_1) = \text{lb}_{SB}(S_2) = \{1\}\), \(\text{ub}_{SB}(S_1) = \text{ub}_{SB}(S_2) = \{1, 2, 3, 4\}\), \(\text{lb}_{C}(S_1) = \text{lb}_{C}(S_2) = \{1\}\), \(\text{ub}_{C}(S_1) = \text{ub}_{C}(S_2) = 2\). \(S_1\) and \(S_2\) are synchronized for all representations. \((\text{minlex} \times \text{card}) + \text{SB}\) fails but \((\text{minlex} \times SB) + \text{card}\) does not prune. \(\square\)

7 Applying the Framework

In the following, we show where the existing combined representations are precisely situated in our framework. More interestingly, we present new results regarding these representations, which help us understand better their pruning capabilities and how they compare to the similar representations that can be constructed using the framework.

Combination of two representations

The \(SB + \text{card}\) representation was used first in set constraint programming in Oz (Müller 2001) and later for instance in the Cardinal set solver (Azvedo 2007). The representation \(SB \times \text{card}\) has been previously referred to as \(\text{subset-cardinality}\) (Malitsky, Sellmann, and van Hoeve 2008) or \(sb\text{-domain}\) (Yip and Hentenryck 2010). A similar BC definition for \(SB \times \text{card}\) is given in (Yip and Hentenryck 2010).

However, there was no formal study nor implementation of \(SB \times \text{card}\).

Although the \(\text{lengthlex}\) representation uses both cardinality and lexicographic information, it can be defined in our framework by a single representation \(T\) defined by a total order. Given that both \(\text{lengthlex}\) and combinations of \(\text{minlex}\) and \(\text{card}\) representations focus on the same information, we might wonder how they compare in general. Interestingly, the \(\text{lengthlex}\) representation is incomparable to representations which deal with the cardinality and lexicographic ordering information either separately or simultaneously.

Theorem 9 \(\text{lengthlex}\) is incomparable to \(\text{minlex}\), \(\text{minlex} + \text{card}\) and \(\text{minlex} \times \text{card}\).

Proof: We first show \(\text{lengthlex} \not\subseteq \text{minlex}\). Consider a conjunction of a lexicographic ordering and two cardinality constraints \((S_1 \preceq_{ML} S_2 \land |S_1| = 2 \land |S_2| = 2)\). Suppose \(D_{ML}(S_1) = \{(2), \{2, 3\}, \{2, 3, 4\}\), \(D_{ML}(S_2) = \{(1, 2), \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3\}\), \(D_{LL}(S_1) = \{(2), \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 3, 4\}, \{2, 4\}, \{3, 4\}, \{1, 2, 3\}\). \(S_1\) and \(S_2\) are synchronized for all representations.

On \(\text{lengthlex}\) fails for both domains but \(\text{lengthlex}\) does not: \(D_{LL}(S_1) = \{(1, 2), \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\) and \(D_{LL}(S_2) = \{(1, 3), \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\) are BC. Thus, \(\text{lengthlex} \not\subseteq \text{minlex}\). and obviously we deduce \(\text{lengthlex} \not\subseteq (\text{minlex} + \text{card})\) and \(\text{lengthlex} \not\subseteq (\text{minlex} \times \text{card})\).

We now show \((\text{minlex} \times \text{card}) \not\subseteq \text{lengthlex}\). Take again the constraint \((S_1 \preceq_{ML} S_2 \land |S_1| = 2 \land |S_2| = 2)\) and \(U = \{1, 2, 3, 4\}\). Suppose \(D_{ML}(S_1) = \{(1, 2, 3, 4), \{1, 2, 4\}, \{1, 3, 4\}, \{1, 4\}\)

Figure 3: Comparison of \(\text{minlex}\) and \(\text{card}\) combinations with \(\text{lengthlex}\): \(H_1 \rightarrow H_2\) means \(H_1 \succ H_2\) whilst \(H_1 - H_2\) means that \(H_1 \sim H_2\).

Combinations of three representations

(Malitsky, Sellmann, and van Hoeve 2008) introduces the \(\text{lengthlex}^*\) representation, which is \(\text{lengthlex} \times SB\) in our framework. In (Yip and Hentenryck 2010), this is called the \(ls\text{-domain}\), but experiments were only performed with \(\text{lengthlex} + \text{SB}\). There was no formal study nor implementation of \(\text{lengthlex} \times SB\). As \(\text{lengthlex} \times \text{SB}\) and combinations of the \(\text{minlex}, \text{SB}\) and \(\text{card}\) represent similar information, we might ask how they compare in general. In Figure 4, we compare the various combinations of \(\text{minlex}\), \(\text{SB}\) and \(\text{card}\) against \(\text{lengthlex} \times \text{SB}\). \(\text{lengthlex} \times \text{SB}\) is incomparable to any \(\text{minlex}\) based combination, including \(\text{minlex}\) alone.

Theorem 10 \(\text{lengthlex} \times \text{SB}\) is incomparable with \(\text{minlex}, \text{minlex} \times \text{SB}, \text{minlex} + \text{card}, \text{minlex} \times \text{card}, \text{minlex} + \text{SB} + \text{card}, (\text{minlex} \times \text{card}) + \text{SB}, (\text{minlex} \times \text{SB}) + \text{card}, (\text{SB} \times \text{card}) + \text{minlex}\) and \(\text{minlex} \times \text{SB} \times \text{card}\).

Proof: First we show that \((\text{lengthlex} \times \text{SB}) \not\subseteq \text{minlex}\). Take the constraint \(S \preceq_{ML} \{1, 3\}\) on the variable \(S\) and \(U = \{1, 2, 3\}\). Suppose \(D_{ML}(S) = \{(1), \{2\}, \{1, 2\}, \{1, 2, 3\}\), \(D_{LL}(S) = \{(1), \{2\}, \{3\}, \{1, 2\}, \{1, 3\}\).
Figure 4: Comparison of minlex based triplets against lengthlex × SB. $H_1 \sim H_2$ means that $H_1 \sim H_2$. 

\{(2, 3), \{1, 2, 3\}\}, \text{bs}_{SB}(S) = \emptyset, \text{ub}_{SB}(S) = \{1, 2, 3\}. \ S_1\ is\ synthesized\ for\ all\ representations.\ Enforcing\ BC\ for\ minlex\ fails\ whilst\ for\ lengthlex × SB\ it\ does\ not.\ Thus\ lengthlex × SB\ is\ not\ stronger\ than\ any\ combination\ of\ minlex.

We now show that \( (\text{minlex × SB × card}) \neq (\text{lengthlex} × \text{SB}) \). Take the constraint \( S_1 \cup S_2 = \{1, 2, 3, 4\} \) on two variables \( S_1 \) and \( S_2 \) and \( U = \{1, 2, 3, 4\} \). Suppose \( D_{LL}(S_1) = D_{LL}(S_2) = \{\{3\}, \{4\}, \{1, 2\}\}, \text{bs}_{SB}(S_1) = \text{bs}_{SB}(S_2) = \{1, 2, 3, 4\} \), \( D_{ML}(S_1) = D_{ML}(S_2) = \{\{1, 2\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3\}, \{1, 3, 4\}, \{1, 4\}, \{2\}, \{2, 3\}, \{2, 3, 4\}, \{2, 4\}, \{3\}, \{3, 4\}, \{4\}\}, \text{ub}_{C}(S_1) = \text{ub}_{C}(S_2) = 1, \text{ub}_{SB}(S_1) = \text{ub}_{SB}(S_2) = 2 \). \( S_1 \) and \( S_2 \) are synthesized for all representations. lengthlex × SB fails whilst minlex × card × SB does not (note that the lower bound of card is updated to 2). Thus all minlex based combinations with SB and card are not stronger than lengthlex × SB. ∎

In (Sadler and Gervet 2008), a combination of maxlex, SB and card, called a hybrid domain, is introduced and associated inference rules are given to maintain consistency between the bounds as well as to enforce some form of local consistency on common set constraints. This hybrid domain is a representation between maxlex + SB + card and maxlex × SB × card. In the hybrid domain, each bound of two representations is updated by looking at the bounds of the other two representations at the same time. This is stronger than reasoning pairwise as we do in maxlex + SB + card. However, this reasoning does not consider the bounds of all three representations simultaneously. Consider for instance a variable \( S \) with the domain \( \text{maxlex}(S) = \{4, 2, 1\}, \text{maxlex}(S) = \{4, 3, 1\}, \text{bs}_{SB}(S) = \{4\}, \text{bs}_{SB}(S) = \{1, 2, 3, 4\}, \text{bc}(S) = \text{ub}_c(S) = 3 \). The inference rules for hybrid domains will not change the subset bounds, but must be in \( \text{bs}_{SB} \) for the bounds to be synchronized. Similarly, consider a variable \( S \) with the domain \( \text{maxlex}(S) = \{4, 3\}, \text{maxlex}(S) = \{5, 1\}, \text{bs}_{SB}(S) = \emptyset, \text{bs}_{SB}(S) = \{1, 2, 3, 4, 5\}, \text{bc}(S) = \text{ub}_c(S) = 2 \). The inference rules will not change the subset bounds, but 2 must be removed from \( \text{bs}_{SB}(S) \) for the bounds to be synchronized. Thus, the hybrid domain is between a weak and strong combination.

**Combination of four representations**

(Law, Lee, and Woo 2009) proposes the combined representation for multiset variables of \( SB + card + variet y \), where variety counts the number of distinct elements \(|a|\). Similar to card, in variety we have \( D_v(S) = \{s \mid \text{lb}_v(S) \leq \text{ub}_v(S) \} \) and given a set of multisets \( A \subseteq 2^U \), we define \( g\text{lb}_v(A) = \min\{\{|a|, a \in A\} \} \) and \( \text{ub}_v(A) = \max\{\{|a|, a \in A\} \} \). In (Law et al. 2011), this representation is extended by considering 8 different lexicographical orderings on multisets. The authors compared these totally ordered representations in terms of expressiveness and compactness, as well as experimentally with respect to \( SB + card + variety \). Our framework lets us combine \( SB, card, variety \) and these other totally ordered representations in new ways, as well as compare their pruning power.

### 8 Complexity Results

Despite its benefits, combining representations can be challenging even with just two representations. For instance, synchronization is intractable to achieve in general. As another example, enforcing BC even on unary constraints can be hard.

**Theorem 11 (Synchronization)** Enforcing synchronization of \( S \) for \( R_1 \) and \( R_2 \) is NP-hard even if \( R_1 \) and \( R_2 \) are defined by total orderings for which computing the successor of an element is polynomial.

**Proof:** Reduction from the NP-hard problem of deciding whether the overlapping alldifferent constraint has a solution (Kutz et al. 2008). Overlapping alldifferent is a conjunction of two alldifferent constraints involving integer variables that overlap on some of these variables. An integer variable \( X_i \) is a variable whose domain \( D(X_i) \) is a finite set of integers. Suppose we have the integer variables \( X_1, \ldots, X_n \) and two constraints alldifferent\( \{X_1, \ldots, X_p\} \) and alldifferent\( \{X_k, \ldots, X_n\} \) with \( k < p \). We build the universe \( U \) so that each occurrence of value \( j \) in a domain is associated with a distinct element: \( U = \{v_j \mid 1 \leq i \leq n, j \in D(X_i)\} \). We define the total ordering \( U \) on \( U \) such that \( v_{i'} < v_{j'} \) if and only if \( i < i' \) or \( i = i' \) and \( j < j' \).

We define \( C_n \) as the subset of \( 2^U \) that contains all sets of size \( n \). We define \( \text{Sol}_1 \) as the subset of \( C_n \) that contains sets \( \{v_{j_1}, \ldots, v_{j_p}, \ldots, v_{j_p}\} \) such that \( j_1, \ldots, j_p \) is a solution of alldifferent\( \{X_1, \ldots, X_p\} \). Similarly, \( \text{Sol}_2 \) is the subset of \( C_n \) that contains sets \( \{v_{j_k}, \ldots, v_{j_k}, \ldots, v_{j_k}\} \) such that \( j_k, \ldots, j_k \) is a solution of alldifferent\( \{X_k, \ldots, X_n\} \). Let \( C_{even} \) be the set of all sets in \( 2^U \setminus C_n \) of even cardinality and let \( C_{odd} \) be the set of all sets in \( 2^U \setminus C_n \) of odd cardinality.

Representation \( R_1 \) is a total ordering \( \leq_{R_1} \) on \( 2^U \) defined as follows. Given two sets \( s_1 \) and \( s_2 \), \( s_1 \leq_{R_1} s_2 \) if and only if \( s_1 \subseteq \text{Sol}_1 \cup C_{even} \) and \( s_2 \not\subseteq \text{Sol}_1 \cup C_{even} \), or \( s_1 \) and \( s_2 \) both belong to \( \text{Sol}_1 \cup C_{even} \) or are both outside \( \text{Sol}_1 \cup C_{even} \) and \( s_1 \leq_{ML} s_2 \). Representation \( R_2 \) is a total ordering \( \leq_{R_2} \) on \( 2^U \) defined as follows. Given two sets \( s_1 \) and \( s_2 \), \( s_1 \leq_{R_2} s_2 \) if and only if \( s_1 \subseteq \text{Sol}_2 \cup C_{odd} \) and \( s_2 \not\subseteq \text{Sol}_2 \cup C_{odd} \), or \( s_2 \) both belong to \( \text{Sol}_2 \cup C_{odd} \) or are both outside \( \text{Sol}_2 \cup C_{odd} \) and \( s_1 \leq_{ML} s_2 \). We observe that computing the successor of a set is a polynomial both in \( R_1 \) and \( R_2 \) because going from one solution of an alldifferent constraint to another in lexicographic order is polynomial.

Suppose now that the set variable \( S \) is represented by \( \text{lb}_{R_1}(S) = \emptyset, \text{ub}_{R_1}(S) = \{v_{j'}, v_{j''}\}, \text{lb}_{R_2}(S) = \{v_{j'}\}, \text{ub}_{R_2}(S) = \{v_{j''}\} \), and \( \text{ub}_{R_1}(S) = \{v_{j'}\} \). \( \text{lb}_{R_1}(S) \) and \( \text{ub}_{R_1}(S) \) both belong
to \( C_{\text{even}} \). Now, for any \( s \in \text{Sol}_1 \) we have \( \text{lb}_{R_1}(S) <_{MB} \text{ub}_{R_1}(S) \). Thus, for any \( s \in \text{Sol}_1 \) we have \( \text{lb}_{R_1}(S) <_{MB} \) \( s <_{RL} \text{ub}_{R_1}(S) \). In addition, for any \( s \notin \text{Sol}_1 \cup C_{\text{even}} \) we have \( \text{ub}_{R_1}(S) <_{RL} s \). As a result, \( \text{Sol}_1 \subseteq D_{R_1}(S) \subseteq \text{Sol}_1 \cup C_{\text{even}} \). Similarly, we prove that \( \text{Sol}_2 \subseteq D_{R_2}(S) \subseteq \text{Sol}_2 \cup C_{\text{odd}} \). The only possible sets common to \( D_{R_1}(S) \) and \( D_{R_2}(S) \) are in \( \text{Sol}_1 \cap \text{Sol}_2 \) because \( C_{\text{even}} \cap C_{\text{odd}} = \emptyset \), \( \text{Sol}_1 \cap C_{\text{odd}} = \emptyset \), and \( \text{Sol}_2 \cap C_{\text{even}} = \emptyset \). Now, sets in \( \text{Sol}_1 \cap \text{Sol}_2 \) correspond to tuples that satisfy both \( \text{all-different}(X_1, \ldots, X_p) \) and \( \text{all-different}(X_{k+1}, \ldots, X_n) \). Therefore, the synchronization of the bounds of \( S \) for \( R_1 \) and \( R_2 \) will not fail if and only if there exists a solution to the overlapping all-different constraint.

\[ \square \]

**Theorem 12 (BC on a unary constraint for \( SB \times card \))**

There exists a unary constraint \( c \) such that enforcing BC on \( c \) for \( SB \), \( card \) and \( SB + card \) are polynomial but for \( SB \times card \) is NP-hard.

**Proof:** Reduction from 3-SAT. Suppose a 3-SAT formula \( F \) on \( n \) Boolean variables. \( S \) is a set variable that contains positive and negative literals and clauses from \( F \). We say that \( S \) contains a truth assignment if \( S \) contains exactly one literal (positive or negative) for each of the \( n \) variables appearing in \( F \). Let \( c(S) \) be satisfied if \( (S \) does not contain exactly \( n \) literals) or \( (S \) contains a truth assignment that satisfies all clauses in \( S \)). Now, enforcing BC on \( c \) for \( SB \) is polynomial. There are several cases to consider. If \( S \) is ground then we simply check \( c \). If \( S \) has all its literals set but the clauses are not ground, then there are several subcases. In the first subcase, \( S \) contains a truth assignment. If some clause in \( \text{lb}(S) \) is not satisfied we fail. Else, we prune from \( \text{ub}(S) \) all clauses that are not satisfied. In the second subcase, \( S \) contains a number of literals different from \( n \). We then have support for all clauses. In the third subcase, we simply check that \( S \) contains \( n \) literals that are not a truth assignment and fail. Other cases (i.e., when not all literals are set) are polynomial by similar arguments. Similarly BC on \( c \) for \( card \) bounds is polynomial as every cardinality has support. Thus, BC on \( SB + card \) is polynomial. However, enforcing BC on \( SB \times card \) is NP-hard. We set \( S \) such that its upper and lower bound include the \( m \) clauses in \( F \), its lower bound is set to include no literals, its upper bound is set to include all \( 2n \) possible literals, and the cardinality of \( S \) is set to \( n + m \). Enforcing BC on \( c \) for \( SB \times card \) will fail unless there is a satisfying assignment to \( F \). \[ \square \]

Apart from these general hardness results regarding combining representations, there also exists the issue of difficulty in reaching fix point in constraint propagation (Sellmann 2009). Our framework helps us right here to give more precise results on specific cases. For instance, consider the \( SB \) and \( card \) representations used in most constraint solvers. When combining these representations using either + or \( \times \), we can only make polynomial number of domain changes before reaching fix point, because only a polynomial number of iterations can be made on \( SB \) and \( card \) bounds, as opposed to for instance \( \text{minlex} \) bounds. Despite the general hardness result given in Theorem 12 on propagating \( SB \times card \), the extra inference needed for BC on \( SB \times card \) on top of BC on \( SB + card \) is polynomial for many common constraints. For reasons of space, we give examples for \( =, \neq \), and \( \subseteq \) but similar results hold for \( \cap, \cup, \) and \( \emptyset \).

For \( S_1 = S_2 \), \( SB \times card \) and \( SB + card \) are equivalent so we need to do nothing more. For \( S_2 \neq S_2 \), the only additional pruning needed over \( SB + card \) is when one set is ground and has cardinality 1, which can be captured by a simple rule. For \( S_1 \subseteq S_2 \), the extra pruning over \( SB + card \) can be captured by the following rules.

1. **Fail:** \( \text{lb}_{C}(S_1) + |\text{ub}_{SB}(S_2) \setminus \text{ub}_{SB}(S_1)| > \text{ub}_{C}(S_2) \)
2. Pruning \( \text{ub}_{SB}(S_2) \) and \( \text{lb}_{SB}(S_1) \):
   - if \( \text{lb}_{C}(S_1) + |\text{ub}_{SB}(S_2) \setminus \text{ub}_{SB}(S_1)| = \text{ub}_{C}(S_2) \) then:
     - \( \forall e \in \text{ub}_{SB}(S_2) \setminus \text{ub}_{SB}(S_1), \) add \( e \) to \( \text{ub}_{SB}(S_1) \)
     - \( \forall e \in \text{ub}_{SB}(S_2) \setminus e \notin \text{lb}_{SB}(S_2) \cup \text{ub}_{SB}(S_1), \) remove \( e \) from \( \text{ub}_{SB}(S_2) \)
3. Pruning \( \text{ub}_{C}(S_1) \):
   - if \( \text{ub}_{SB}(S_2) \leq \text{ub}_{C}(S_2) - |\text{ub}_{SB}(S_2) \cap \text{ub}_{SB}(S_1)| \)
4. Pruning \( \text{lb}_{C}(S_2) \):
   - if \( \text{lb}_{SB}(S_2) \leq \text{lb}_{C}(S_2) - |\text{lb}_{SB}(S_2) \cap \text{ub}_{SB}(S_1)| \)

No need to modify \( \text{lb}_{C}(S_1), \text{ub}_{C}(S_2), \text{ub}_{SB}(S_1), \text{lb}_{SB}(S_2) \).

9 Experimental Results

To compare combinations of representations which are theoretically incomparable, we ran some experiments. We select \( \text{minlex} \) (\( ML \)), \( \text{lengthlex} \) (\( LL \)), \( SB \), \( card \) (\( C \)), and their combinations, and evaluate the information lost when transforming a domain from one of these representations to another representation. The goal is to predict how much the pruning performed by BC on one representation propagates to the others. Given a pair of (combined) representations \( R_1 \) and \( R_2 \), we randomly generate the bounds of a set \( S \) in \( R_1 \), initialize the bounds of \( S \) in \( R_2 \) so that \( D_{R_2}(S) = 2^m \), and then synchronize the bounds of \( S \) in \( R_2 \) with the bounds of \( S \) in \( R_1 \). Finally we compute the approximation ratio \( D_{R_2}(S) / D_{R_1}(S) \). We average over 1000 sets. This gives us the entry \((R_1, R_2)\) in Table 1.

We report results for \( |U| = 10 \). However, similar observations hold for larger universes. As we indicate in bold, 4 out of 9 representations have mean approximation ratio less than 3. These representations, in order of increasing mean approximation ratio, are \( ML \times SB \times C \), \( LL \times SB \), \( ML \times SB \), and \( SB \times C \). Surprisingly, all these representations include the subset bounds. The remaining representations are either \( SB \) or do not include subset bounds, and exhibit much looser approximations. In particular, a relatively simple combination like \( SB \times C \) behaves much better than for instance \( LL \). The best binary combination \( LL \times SB \) significantly improves \( LL \).

10 Related Work

In his Ph.D. thesis (Jefferson 2007), Jefferson has provided a framework for comparing set representations and their combinations. For instance he presented a result closely related to Theorem 12 though he obtained it via a different way.
Differently from (Jefferson 2007), our framework allows the study of many different ways of combining multiple representations most of which were not previously known. In addition, the ordering we have defined on representations lets us compare the pruning power of the (combined) representations and consequently measure their ability to prune the search space. (Jefferson 2007) instead discusses search size comparison only briefly by showing representations which can represent more and produce smaller search trees. Altogether, in our work we can see what kind of combinations can be built, how they compare and which ones already exist or not.

11 Conclusions and Future Work

We have defined a general framework for combining together different representations of set and multiset variables. Central to this framework is the idea of synchronization between representations. We have proposed two different combinators, a weak method which considers each representation independently, and a strong method which considers them together. To compare representations, we have proposed an ordering on representations that measures their ability to prune the search space. We have undertaken a detailed study of combinations of representations (including several like \textit{SB} \times \textit{card} and \textit{lengthlex} \times \textit{SB} which have been proposed previously but not compared theoretically). We identified several new relations. For example, the \textit{lengthlex} representation, which simultaneously captures cardinality and lexicographic ordering information, is not better than a representation which deals with cardinality and lexicographic ordering information separately or simultaneously. As a second example, the hybrid domain of (Sadler and Gervet 2008) provides between the weak and strong combination of the constituent representations. Finally, we have provided some complexity results regarding synchronization and propagation, and performed experiments showing how good or bad a representation is in approximating others.

This paper opens the door to several interesting new directions. First, we believe that our framework along with the complexity results provide useful step towards focusing on particular combinations with promising theoretical properties and conforming their interest on practical problems. Second, our theoretical and experimental results suggest we should develop solvers that more tightly integrate subset bounds with representations like cardinality and \textit{lengthlex}. All these necessitate the design and development of efficient synchronization and propagation algorithms. Last, our framework helps us to give more precise results on the reaching of fixed points when combining representations (Sellmann 2009). An in-depth investigation is needed to understand how certain combinations impact on this issue.

References


Table 1: Comparison of approximation ratios of different (combined) representations.

| $|U| = 10$ | $ML$ | $LL$ | $SB$ | $ML \times C$ | $ML \times LL$ | $ML \times SB$ | $LL \times SB$ | $SB \times C$ | $ML \times SB \times C$ |
|---------|------|------|------|-------------|-------------|-------------|-------------|-------------|-------------|
| Original Representation | ML   | LL   | SB   | ML $\times$ C | ML $\times$ LL | ML $\times$ SB | LL $\times$ SB | SB $\times$ C | ML $\times$ SB $\times$ C |
| ML      | 1    | 1    | 2.41 | 1            | 1            | 2.36         | 2.38         | 1            | 2.38         |
| LL      | 5.03 | 1    | 1.97 | 4.82         | 1            | 2.27         | 1.97         | 1            | 2.27         |
| SB      | 19.8 | 30.7 | 1    | 17.6         | 16.1         | 11.7         | 1            | 1            | 1            |
| ML $\times$ C | 2.39 | 5.03 | 1    | 1            | 2.28         | 1            | 1            | 1            | 1            |
| ML $\times$ LL | 41.2 | 24   | 5.03 | 1            | 5.1          | 1            | 2.42         | 2.42         | 1            |
| ML $\times$ SB | 30.6 | 24   | 1    | 17.6         | 17.6         | 13.7         | 1            | 1            | 1            |
| LL $\times$ SB | 48.5 | 30.6 | 1    | 17.6         | 1            | 1            | 2.42         | 2.42         | 1            |
| SB $\times$ C | 27.9 | 24   | 1    | 13.7         | 1            | 1            | 1            | 1            | 1            |
| ML $\times$ SB $\times$ C | 38.1 | 30.6 | 4.01 | 1            | 13.7         | 1            | 2.42         | 2.42         | 1            |

Geometric mean | 13.6 | 13.7 | 3.42 | 7.26 | 3.98 | 1.87 | 1.54 | 2.11 | 1.3 |


