Combinations of theories and the Bernays-Schönfinkel-Ramsey class

Pascal Fontaine

LORIA, Nancy University, France

Abstract. The Bernays-Schönfinkel-Ramsey (BSR) class of formulas is the class of formulas that, when written in prenex normal form, have an $\exists^*\forall^*$ quantifier prefix and do not contain any function symbols. This class is decidable. We show here that BSR theories can furthermore be combined with another disjoint decidable theory, so that we obtain a decision procedure for quantifier-free formulas in the combination of the BSR theory and another decidable theory.
The classical Nelson-Oppen combination scheme requires theories to be stably-infinite, ensuring that, if a model is found for both theories in the combination, models agree on cardinalities and a global model can be built. We show that combinations with BSR theories can be much more permissive, even though BSR theories are not always stably-infinite. We state that it is possible to describe exactly all the (finite or infinite) cardinalities of the models of a given BSR theory. For the other theory, it is thus only required to be able to decide if there exists a model of a given cardinality.
With this result, it is notably possible to use some set operators, operators on relations, orders — any operator that can be expressed by a set of BSR formulas — together with the usual objects of SMT solvers, notably integers, reals, uninterpreted symbols, enumerated types.

1 Introduction

Many techniques for the formal verification of information systems generate *verification conditions*, i.e. formulas encapsulating parts of the reasoning about the systems. The deduction tools validating these verification conditions should accept expressive languages, and should require a minimal amount of human interaction. Combination of theories is the method behind SMT-solvers (SMT for satisfiability modulo theories) to build decision procedures for very expressive languages, containing interpreted symbols from several decidable theories. Usually the theory embedded in the solvers is a static combination of linear arithmetic, uninterpreted symbols, list operators, bit-vectors,... For instance, it is possible to combine a decision procedure for integer linear arithmetic and a decision procedure for the empty theory (i.e. a decision procedure for equality and uninterpreted symbols) into a decision procedure to study formulas like

$$x \leq y \wedge y \leq x + f(x) \wedge P(h(x) - h(y)) \wedge \neg P(0) \wedge f(x) = 0.$$

The Bernays-Schönfinkel-Ramsey (BSR) class is a wide decidable class of formulas; any set of function-free universal formulas is indeed decidable. We consider here this class of formulas as a component in a combination of theories.

The classical Nelson-Oppen combination scheme [11, 16] requires every theory in the combination to be stably-infinite, i.e. every quantifier-free formula satisfiable in the theory should have a model with infinite cardinality. BSR theories are not, in general, stably-infinite: as an example, consider the BSR theory $\forall x \forall y (x = y)$ that only accepts models on a domain with one element. The classical combination result is not suitable in our case.

It has already been mentioned [17] that a BSR theory can be combined with a theory $\mathcal{T}$ provided

- if a set of ground literals L is $\mathcal{T}$-satisfiable, then the minimal cardinality of $\mathcal{T}$-models for L can be computed;
- $\mathcal{T}$ only has finite models.

We show here that this last strong requirement is not necessary; BSR theories can in fact be combined with any other decidable theory $\mathcal{T}$ (with or without infinite models, stably infinite or not), provided that, if a set L of ground literals is satisfiable in $\mathcal{T}$, it is possible to determine if there exists a $\mathcal{T}$-model of a given finite or infinite cardinality.

Motivations: the incentive for the procedure we present in Section 6 is double. First, the requirement we impose on the theory $\mathcal{T}$ is fulfilled by many decidable theories; using results in this paper it is possible to extend many decidable quantifier-languages (for instance, mixing uninterpreted symbols with linear arithmetic on reals and integers) with new interpreted predicates defined by a BSR theory. The BSR theory is not required to be stably-infinite. The other theory is not required to have only finite models.

The second motivation for such a general combination of theories is that the $\mathcal{T}$-satisfiability of quantifier-free formulas containing operators on sets, relations,... can be reduced to studying the satisfiability of sets of literals in the combinations of $\mathcal{T}$ and a BSR theory (see Sections 3 and 4). In Section 5, we show that there is a straightforward implementation of this method when $\mathcal{T}$ is the empty theory. Good results have been obtained with our prototype on translations of some problems from the SET domain of the TPTP library. When $\mathcal{T}$ is not the empty theory, we can fall back to the general decision procedure in Section 6. This decision procedure relies on the computation of model cardinalities of BSR theories. We show in Section 7 that it is possible to know exactly the cardinalities of BSR theories, and, in particular, we prove that it is possible to compute if a BSR theory has an infinite model or not.

For convenience, the results in this paper are presented in an *unsorted* framework, although most SMT-solvers work on a many-sorted logic (see for instance [5]). The results can easily be transferred to a many-sorted framework, at an expense of heavier notations.

2 Notations

A first-order language is a tuple $\mathcal{L} = \langle \mathcal{V}, \mathcal{F}, \mathcal{P} \rangle$ such that $\mathcal{V}$ is a enumerable set of variables, $\mathcal{F}$ and $\mathcal{P}$ are sets of functions and predicates (we refer to "symbols" for the union of $\mathcal{F}$ and $\mathcal{P}$). Every function and predicate symbol is assigned an arity. Nullary predicates are propositions, and nullary functions are constants. The set of terms on language $\mathcal{L}$ is defined in the usual way. A ground term is a term without variables. An atomic formula is either $t = t'$ where t and t' are terms, or a predicate symbol applied to the right number of terms. Formulas are built from atomic formulas, connectors ($\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\equiv$), and quantifiers ($\forall$, $\exists$). A formula with no free variable is closed. A theory is a set of closed formulas. Two theories are disjoint if no predicate (except the equality) or function symbol is interpreted in both theories.

An interpretation $\mathcal{I}$ for a first-order language assigns a set of elements D to the domain, a total function $\mathcal{I}[f]$ on D with appropriate arity to every function symbol f, a predicate $\mathcal{I}[p]$ on D with appropriate arity to every predicate symbol p, and an element $\mathcal{I}[x]$ to every variable x. By extension, an interpretation gives a value in D to every term, and a truth value to every formula. A model for a formula (or a theory) is an interpretation that makes the formula (resp. every formula in the theory) true. A formula is satisfiable if it has a model. It is unsatisfiable otherwise. A formula G is $\mathcal{T}$-satisfiable if it satisfiable in the theory $\mathcal{T}$, that is, if $\mathcal{T} \cup \{G\}$ is satisfiable. A $\mathcal{T}$-model of G is a model of $\mathcal{T} \cup \{G\}$. A formula G is $\mathcal{T}$-unsatisfiable if it has no $\mathcal{T}$-model.

The cardinality of an interpretation (or model) is the cardinality of the domain of this interpretation. The restriction of a predicate p on domain D to domain $D' \subseteq D$ is the predicate p' with domain D' such that p and p' have the same truth value for all arguments in D'.

A conjunctive (disjunctive) normal form is a conjunction of clauses, i.e. a conjunction of disjunctions of literals, (resp. a disjunction of conjunctions of literals). It is always possible to transform a quantifier-free formula into a logically equivalent conjunctive (disjunctive) normal form. A formula is universal if it is of the form $\forall x_1 \ldots \forall x_n . \varphi$ where φ is quantifier-free. A Skolem formula is a formula where all universal quantifiers appear with a positive polarity only, and all existential quantifiers appear with a negative polarity only. It is always possible to transform a given formula into an equisatisfiable Skolem formula, using Skolemization. We refer to [3] for Skolemization and conjunctive (disjunctive) normal form transformations.

3 From operators to BSR theories

Objects such as sets, relations, or arrays of bits can be viewed as predicates. For instance, sets can be unambiguously represented by their characteristic function

Equality	$\approx$	$\lambda p\, q.\ \forall x.\ p(x) \equiv q(x)$
membership	$\in$	$\lambda x\, p.\ p(x)$
$\emptyset$	$\emptyset$	$\lambda x.\ \bot$
Ω	Ω	$\lambda x.\ \top$
Enumerate	$\{a_1, \ldots a_n\}$	$\lambda x. (x = a_1 \vee \ldots x = a_n)$
Intersection	$\cap$	$\lambda p\, q.\ \lambda x.\ p(x) \wedge q(x)$
Union	$\cup$	$\lambda p\, q.\ \lambda x.\ p(x) \vee q(x)$
Difference	$\setminus$	$\lambda p\, q.\ \lambda x.\ p(x) \wedge \neg q(x)$
Subset	$\subseteq$	$\lambda p\, q.\ \forall x.\ p(x) \Rightarrow q(x)$

A. Sets

Equality	$\approx$	$\lambda p\, q.\ \forall x\, y\ .\ p(x,y) \equiv q(x,y)$
Transitive	Trans	$\lambda p.\ \forall x\, y\, z\ .\ [p(x,y) \wedge p(y,z)] \Rightarrow p(x,z)$
Symmetric	Sym	$\lambda p.\ \forall x\, y\ .\ p(x,y) \equiv p(y,x)$
Antisym.	ASym	$\lambda p.\ \forall x\, y\ .\ \neg p(x,y) \vee \neg p(y,x) \vee x = y$
Total	Tot	$\lambda p.\ \forall x\, y\ .\ p(x,y) \vee p(y,x)$
Reflexive	Refl	$\lambda p.\ \forall x\ p(x,x)$
Irreflexive	ARefl	$\lambda p.\ \forall x\ \neg p(x,x)$
Identity	Id	$\lambda x\, y.\ x = y$
Product	$\times$	$\lambda p\, q.\ \lambda x\, y.\ p(x) \wedge q(y)$

B. Relations

Equality	$\approx$	$\lambda p\, q.\ \forall x.\ p(x) \equiv q(x)$
Reading	read	$\lambda p\, i.\ p(i)$
Writing	write	$\lambda p\, i\, x.\ \lambda j.\ (j = i \Rightarrow x) \wedge (j \neq i \Rightarrow p(j))$

C. One-dimensional arrays of bits

Fig. 1. Operators

and operators on sets can be viewed as operators on predicates. In Figure 1, we give a few examples of set-like operators, operators on relations, operators to encode read and write operations on arrays of bits. In those examples, we assume p and q are predicates of appropriate arity and x, y, z are (first-order) variables. Notice that set-like operators can also be defined for relations; for instance, the intersection of relations is defined as $\lambda p\, q.\ \lambda x, y.\ p(x,y) \wedge q(x,y)$.

We consider formulas that are written in a first-order language augmented with the operators — defined as λ-terms given in Figure 1 — applied to the right number of objects of appropriate type.

Example 1. if A, B, C are unary predicates used to represent sets, a formula may contain $A \approx B \cap C$ which becomes, after substitution of $\cap$ and $\approx$ by their definition

$$[\lambda p\, q.\ \forall x.\ p(x) \equiv q(x)]\, (A, (\lambda p\, q.\ \lambda x.\ p(x) \wedge q(x))(B, C))\,.$$

After β-reduction, this becomes

$$\forall x. A(x) \equiv [B(x) \wedge C(x)]\,. \tag{1}$$

In general, the formulas obtained after elimination of operators mentioned in this section are first-order, but may contain quantifiers. Those quantifiers come directly from the λ-terms; for instance the quantifier in (1) comes from the definition of $\approx$. It is easily shown however, that, if the original formula (with operators on sets, relations…) does not contain quantifiers, the resulting first-order formula is a Boolean combination of (atoms and) formulas of the form $\forall x_1 \dots x_n\, \varphi$ where φ is quantifier-free. Furthermore, quantified variables are used only as arguments of predicates, that is, no function has a quantified variable as an argument.

4 From FOL formulas to combination of theories

The formulas obtained in the previous section are Boolean combinations of quantified formulas. In this section we describe the process to reduce the $\mathcal{T}$-satisfiability problem of these quantified formulas, to the satisfiability problem for sets of literals in the union of two theories: $\mathcal{T}$ and a disjoint Bernays-Schönfinkel-Ramsey theory $L_\forall$. For the rest of the paper, we only impose one restriction on the decidable theory $\mathcal{T}$: if a set of literals is $\mathcal{T}$-satisfiable, it is possible to compute if there exists a model of a given cardinality. We also assume that all predicates occurring in operators from Figure 1 are uninterpreted for $\mathcal{T}$.

The form of the formulas issued in the previous section is such that a structural Skolemization (see for instance [3]) will never introduce Skolem functions, but only Skolem constants. We assume that the formula is Skolemized, using such a structural Skolemization. The obtained formula is a Boolean combination of universal formulas (and atoms), the universal formulas appearing with a positive polarity only.

The usual technique used in SMT-solvers to check the satisfiability of a quantifier-free formula in a theory $\mathcal{T}$ is a (loose or tight) cooperation of a Boolean satisfiability checker, and a procedure to check the satisfiability of literals within $\mathcal{T}$. This cooperation splits the problem into two parts: first, pure Boolean model searches, and second, $\mathcal{T}$-satisfiability checks for the corresponding *conjunctive sets of literals*. For simplicity, we consider here that the split is realized by converting the formula to disjunctive normal form. The formula is satisfiable if and only if at least one conjunction of literals in the disjunctive normal form is satisfiable. Now assume Ψ is the obtained formula after Skolemization. The formula is transformed into disjunctive normal form, the quantified parts being left unchanged. Since the formula has been Skolemized, the remaining (universal) quantifiers all appear with a positive polarity. Each conjunction of literals in the disjunctive normal form only contains:

- first-order literals;
- formulas of the form $\forall x_1 \dots x_n\, \varphi$, where φ is a quantifier-free formula, such that no $x_1 \dots x_n$ is used within a function;

Example 2. Suppose we want to study the satisfiability of the formula:

$$a = b \wedge f(a) \in A \wedge f(b) \notin C \wedge [f(b) \notin A \vee A \cup B \approx C \cap D] .$$

Substituting operators $\in$, $\cup$, $\cap$, $\approx$ by their definition and applying β-reduction, one obtains

$$a = b \wedge A(f(a)) \wedge \neg C(f(b)) \wedge \Big[\neg A(f(b)) \vee \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\Big]$$

Structural Skolemization leaves this last formula unchanged, since the sole universal quantifier appears with a positive polarity. The corresponding disjunctive normal form contains the two conjunctive sets of literals:

$$\{a = b, A(f(a)), \neg C(f(b)), \neg A(f(b))\} \tag{2}$$
$$\{a = b, A(f(a)), \neg C(f(b)), \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\} \tag{3}$$

The first set can easily be identified as being unsatisfiable. The second set only contains first-order (quantifier-free) literals and formulas of the form $\forall x_1 \dots x_n\, \varphi$, where φ is a quantifier-free formula, such that no $x_1 \dots x_n$ is used within a function.

In order to study the satisfiability of a set of literals in the combination of disjoint theories, one usually first computes a *separation* of the set of formulas along the languages in the disjoint theories.[1] Each part of the separation contains only the symbols from one theory in the combination; the only shared symbols are equality and variables. We apply the same technique to separate predicates that appear in quantified formulas from the rest of the symbols. For instance, the set (3) is logically equivalent (in whatever theory) to the union of the sets

$$L_g = \{a = b, y = f(a), z = f(b)\},$$
$$L_\forall = \{A(y), \neg C(z), \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\},$$

where y, z are introduced variables. In general, a separation can be built using the following method.

Algorithm: Initially, L is a set containing literals and universal formulas, and no quantified variable as argument of a function. The separation algorithm builds two sets L_g (g for ground), and $L_\forall$ (for quantified formulas and related predicates):

- for every uninterpreted predicate p that occurs in a quantified formula in L, for every occurrence $p(t_1, \dots t_n)$ of this predicate (in a quantified formula or not), for every subterm t_i that is not a variable (shared or not), introduce a new shared variable x, add $x = t_i$ to L_g, and replace t_i by x in L. Handle similarly all occurrences of the form $t_1 = t_2$ in a quantified formula in L. This is possible since no quantified variable is used as an argument of a function;

[1] See for instance [6] for a formal presentation of the separation technique.

– for every uninterpreted predicate p that belongs to a quantified formula in L, add every literal $p(t_1, \ldots t_n)$ (or $\neg p(t_1, \ldots t_n)$) from L to $L_\forall$. The previous two steps ensure that here, $t_1, \ldots t_n$ are variables;
– add every quantified formula from L to $L_\forall$. Those formulas are universal formulas, and the previous steps ensures that they are function-free.
– finally, every literal in L that does not belong to $L_\forall$ is added to L_g.

In this algorithm:

– the computed $L_\forall$ is a set of function-free universal formulas, i.e. a BSR theory;
– the initial L is $\mathcal{T}$-satisfiable if and only if $L_g \cup L_\forall$ is also $\mathcal{T}$-satisfiable;
– the shared terms in L_g and $L_\forall$ are all variables.

To summarize, studying the $\mathcal{T}$-satisfiability of a given formula with operators as described in Figure 1 can be reduced to studying the $\mathcal{T}$-satisfiability of sets $L_g \cup L_\forall$. Another point of view is to study the satisfiability of the sets of literals L_g, in the combination of the disjoint theories $\mathcal{T}$ and $L_\forall$. In the following sections, we show that this problem is decidable, for any decidable theory $\mathcal{T}$, as long as it is possible to determine if L_g accepts a $\mathcal{T}$-model of a given cardinality.

5 Combining a BSR theory with the empty theory

The method in the previous section leads to checking the satisfiability of a set of literals L_g in the union of $\mathcal{T}$ and a BSR theory $L_\forall$ (L_g and $L_\forall$ share only variables). We assume in this section that $\mathcal{T}$ is the empty theory. That is, every function and predicate in L_g is left uninterpreted.

The classical Nelson-Oppen combination scheme cannot be used, since the theory $L_\forall$ is not necessarily stably-infinite, that is, it may be satisfiable only in finite models. For instance, if the original formula uses the "Enumerate" operator, the resulting sets of formulas may contain a formula of the form

$$\forall x.\ x = a \vee x = b \vee x = c$$

which would make $L_\forall$ non stably-infinite; the formula accepts models of cardinality at most three. However we known that the empty theory can be combined with any theory, not only stably-infinite ones [7, 17]. We now recall the combination algorithm.

Given a partition $\mathcal{P}$ of a set of terms, an *arrangement* induced by $\mathcal{P}$ is the set of all equalities between any two terms in the same class of $\mathcal{P}$, and all disequalities between any two terms in different classes in $\mathcal{P}$. For instance, the arrangement induced by $\{\{x_1, x_2\}, \{x_3\}\}$ is $\{x_1 = x_2, x_1 \neq x_3, x_2 \neq x_3\}$. Assume we have to study the satisfiability of the separation $L_1 \cup L_2$ in the combination of the stably-infinite disjoint theories $\mathcal{T}_1$ and $\mathcal{T}_2$, where L_i ($i \in \{1, 2\}$) only contains symbols from $\mathcal{T}_i$ and variables. The classical result for combining stably-infinite disjoint theories states that $L_1 \cup L_2$ is satisfiable in the combination of $\mathcal{T}_1$ and $\mathcal{T}_2$ if and only if there exists an arrangement $\mathcal{A}$ on the set of shared variables between L_1 and L_2, such that $L_i \cup \mathcal{A}$ is $\mathcal{T}_i$-satisfiable, for $i = 1$ and $i = 2$. The procedure terminates, since the set of shared variables is finite, as well as the set of arrangements. In the case where $\mathcal{T}_1$ is the empty theory, and $\mathcal{T}_2$ is any theory (not necessarily stably-infinite), the result still holds [7], but the arrangement has to be considered on a larger set of terms; the arrangement has to be considered on all terms and variables in $L_1 \cup L_2$.[2]

Applied to our present case, L_g is satisfiable in the combination of the empty theory and $L_\forall$, if and only if there exists an arrangement $\mathcal{A}$ of all ground terms and free variables in L_g such that $\mathcal{A} \cup L_g$ and $\mathcal{A} \cup L_\forall$ are both satisfiable.

Example 3. As an application, consider again the previous example:

$$\{a = b, A(f(a)), \neg C(f(b)), \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\}$$

one has to study the satisfiability of the unions of the sets

$$L_g = \{a = b, y = f(a), z = f(b)\},$$
$$L_\forall = \{A(y), \neg C(z), \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\},$$

which is equivalent to study the satisfiability of L_g in the combination of the empty theory and $L_\forall$. The combination framework then ensures that $L_g \cup L_\forall$ is satisfiable if and only if there exists an arrangement $\mathcal{A}$ of $\{a, y, z\}$ (the other terms being necessarily equal to one in this set) such that $\mathcal{A} \cup L_g$ and $\mathcal{A} \cup L_\forall$ are satisfiable. There are well known decision procedures for both satisfiability problems.

5.1 Towards an implementation

The set of formulas $\mathcal{A} \cup L_\forall$ is also a BSR theory. It is satisfiable if and only if $\mathcal{A} \cup L_{\text{inst}}$ is, where L_{inst} is a set of well-chosen instances of formulas in $L_\forall$. This leads to the following result:

Theorem 1. *Given a theory $\mathcal{T}$, a set of literals L_g, a BSR theory $L_\forall$ such that L_g and $L_\forall$ only share variables, then $L_g \cup L_\forall$ is satisfiable (in the empty theory) if and only if $L_g \cup L_{\text{inst}}$ is, where L_{inst} is a set of instances of $L_\forall$: for every formula $\forall x_1 \ldots \forall x_n \varphi(x_1, \ldots x_n)$ in $L_\forall$ ($\varphi(x_1, \ldots x_n)$ being quantifier-free), and terms or free variables $t_1, \ldots t_n$ in $L_g \cup L_\forall$, L_{inst} contains the formula $\varphi(t_1, \ldots t_n)$.*

Example 4. Applying this result on the previous example:

$$L_g = \{a = b, y = f(a), z = f(b)\},$$
$$L_\forall = \{A(y), \neg C(z), \forall x.\ [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]\},$$

[2] Another approach considers arrangements on the set of shared variables only, computes minimal cardinalities of models for the empty theory, and ensures there is a model with a larger (or equal) cardinality for the other theory [17].

gives the formula

$$\{a = b, y = f(a), z = f(b), A(y), \neg C(z), \varphi(y), \varphi(z), \varphi(a)\}$$

where $\varphi(x) = [A(x) \vee B(x)] \equiv [C(x) \wedge D(x)]$.

Deciding formulas that only contain operators like those in Figure 1, can be done easily using the capabilities implemented in any SMT-solver (for instance [1, 12]): the ability to deal with a Boolean combination of terms that only contain uninterpreted symbols. This language being handled very efficiently by modern solvers, the tools do cope well even if the number of generated instances is large. A naïve implementation can be realized by doing β-reduction, Skolemization, and instantiation as a preprocess, feeding a Boolean combination of terms with only uninterpreted symbols to the SMT-solver.

A working prototype has been implemented. We ran this prototype on the translation of problems SET008+3p, SET064+1p, SET143+3p, SET171+3p, SET580+3p, SET601+3p SET606+3p, SET623+3p, and SET609+3p from the TPTP library [15]. Unsurprisingly, these are all solved in a few milliseconds. We should however mention that it is not really relevant to compare these performances with ones of the FOL provers, since the set theories in which the problems are checked for satisfiability are not the same for both approaches. For instance, our approach implicitly assume sets cannot contain other sets, whereas no such assumption is made in the TPTP problems.

6 Combining BSR theories with arbitrary decidable theories

In the previous sections we considered formulas that contain uninterpreted symbols, as well as other symbols such as set and relation operators. We show in this section that there is a decision procedure for formulas that contain such set and relation operators and *interpreted* symbols from an arbitrary decidable theory $\mathcal{T}$, provided (1) there is a decision procedure for the arbitrary theory that is able to state if there is a model of a given cardinality (2) set and relation operators are applied on uninterpreted symbols only. We have to study the satisfiability of the set of ground literals L_g in the combination of the disjoint theories $\mathcal{T}$ and $L_\forall$, where L_g only contains symbols from $\mathcal{T}$ and variables.

Theorem 2. *Given a theory $\mathcal{T}$, a set of literals L_g, and a BSR theory $L_\forall$ such that L_g and $L_\forall$ only share variables, then L_g is satisfiable in $\mathcal{T} \cup L_\forall$ if and only if there exists an arrangement $\mathcal{A}$ of variables shared by L_g and $L_\forall$ such that $\mathcal{A} \cup L_g$ has a $\mathcal{T}$-model, and $\mathcal{A} \cup L_\forall$ has a model, both models having the same cardinality.*

This theorem is an adaptation of the general result to combine non-stably-infinite theories (see for instance [17]).

For theoretic discussions, the process of combining stably-infinite theories usually implies guessing an arrangement on a set of variables. In practice, it is equivalent, and more efficient that decision procedures exchange disjunctions of equalities (see for instance [6] for a presentation of this equivalence). We can imagine a similar treatment here for cardinalities. The decision procedures could negotiate the size of the models by exchanging constraints. For simplicity, a naïve decision procedure for the combination can be:

- build L_g and $L_\forall$ according to the method presented in section 4. Both sets only share variables, and no symbol in $L_\forall$ is interpreted by $\mathcal{T}$;
- guess an arrangement $\mathcal{A}$ on shared variables. Notice there is only a finite number of such arrangements: this guess can thus be replaced by a terminating loop;
- if the code on Fig. 2 returns "succeed" for $\mathcal{A}$, the $L_g \cup L_\forall$ is $\mathcal{T}$-satisfiable;
- If every arrangement returns "fail" for the code on Fig. 2, $L_g \cup L_\forall$ is $\mathcal{T}$-unsatisfiable.

The procedure concludes to $\mathcal{T}$-satisfiability if and only if a model is found that meets the conditions of Theorem 2. It remains to check that that every step of the code on Fig. 2 is tractable. The test on line 1 is decidable since the $\mathcal{T}$-satisfiability problem for sets of literals is decidable, and since $\mathcal{A} \cup L_\forall$ is a BSR theory (decidable fragment). The results in the following section state that it is possible to determine exactly what cardinalities are accepted for models of any BSR theory, and in particular for $\mathcal{A} \cup L_\forall$: the tests on lines 3 and 7 are decidable, and it is possible to enumerate (within finite time) the cardinalities in line 4. The tests on lines 5, 8 and 13 are possible thanks to the condition on theory $\mathcal{T}$. For the test on line 15, checking if $\mathcal{A} \cup L_g$ has a $\mathcal{T}$-model with cardinality greater or equal to k is simply reduced to checking the $\mathcal{T}$-satisfiability of $\mathcal{A} \cup L_g \cup \bigcup_{1 \leq i \leq k} \bigcup_{i < j \leq k} \{a_i \neq a_j\}$ where $a_1, \ldots a_k$ are fresh constants.

7 Cardinalities of BSR theories

The previous section states that to combine a BSR theory with another theory is mainly a matter of getting the cardinalities of the models of the BSR theory. We give now a necessary and sufficient criteria to determine if there is an infinite model for such a theory, and if not, what are the finite cardinalities for which there exists a model. For simplicity, we assume here that we have a BSR theory, with no free variables, but only constants. If this requirement is not met, one can transform the problem into an equivalent one by replacing free variables with fresh constants.

if $\mathcal{A} \cup L_g$ is $\mathcal{T}$-unsatisfiable or
  $\mathcal{A} \cup L_\forall$ is unsatisfiable **then**
  return fail
if $\mathcal{A} \cup L_\forall$ only has finite models **then**
  for each cardinality j of models of $\mathcal{A} \cup L_\forall$ **do**
    if $\mathcal{A} \cup L_g$ has a $\mathcal{T}$-model with cardinality j **then**
      return succeed
if $\mathcal{A} \cup L_\forall$ has an infinite model **then**
  if $\mathcal{A} \cup L_g$ has an infinite $\mathcal{T}$-model **then**
    return succeed
  else
    $k :=$ the number of free variables and constants in $\mathcal{A} \cup L_\forall$
    for each $j < k$ **do**
      if $\mathcal{A} \cup L_g$ has a $\mathcal{T}$-model with cardinality j and
        $\mathcal{A} \cup L_\forall$ has a model with cardinality j **then**
        return succeed
    if $\mathcal{A} \cup L_g$ has a $\mathcal{T}$-model with cardinality $\geq k$ **then**
      return succeed
return fail

Fig. 2. Inspecting arrangement $\mathcal{A}$

Given a BSR theory $\mathcal{T}$ using k constants, we first recall the simple result that states that, if $\mathcal{T}$ has a model of (finite or infinite) cardinality i greater than k, then it has a model for every cardinality j such that $k \leq j \leq i$. We then show that there is a number k' ($> k$), computable from $\mathcal{T}$, such that, if there is a model of cardinality greater or equal to k', then there is an infinite model. Altogether, this implies that $\mathcal{T}$ either has a model for every cardinality greater or equal to k (example in Figure 3), or there exists a j smaller than the known, finite, number k', such that $\mathcal{T}$ has a model of every cardinality between k and j, and no model of cardinality greater than j (example in Figure 3). Alternatively, one can also decide if a theory with n distinct quantified variables has an infinite model by checking if it has a n-repetitive model (see subsection 7.2).

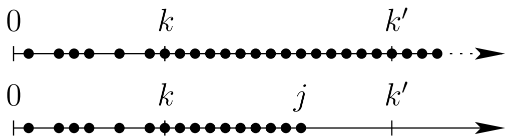

Fig. 3. Theories with infinite (above) and finite cardinalities.
A dot means there is a model with given cardinality.

7.1 BSR theories and finite models

Intuitively, the following theorem states that, given a model for a BSR theory, the elements in the domain that are not assigned to ground terms (i.e. the constants) can be eliminated, keeping it a model:

Theorem 3. *Given a model $\mathcal{M}$ for a BSR theory $\mathcal{T}$ with domain D, then $\mathcal{M}'$ such that*

- *the domain is a non-empty set $D' \subseteq D$, with $\mathcal{M}[a] \in D'$ for every constant a in $\mathcal{T}$;*
- *for every predicate p, $\mathcal{M}'[p]$ is the restriction of $\mathcal{M}[p]$ to the domain D';*

is also a model for $\mathcal{T}$

Proof. Since $\mathcal{M}$ is a model for $\mathcal{T}$, for each closed formula $\forall x_1 \ldots x_n.\varphi$ in $\mathcal{T}$ (where φ is function and quantifier-free), and for all $d_1, \ldots d_n \in D' \subseteq D$, $\mathcal{M}_{x_1/d_1,\ldots x_n/d_n}$ is a model for φ. This also means that, for all $d_1, \ldots d_n \in D'$, $\mathcal{M}'_{x_1/d_1,\ldots x_n/d_n}$ is a model for φ, and finally that $\mathcal{M}'$ is a model for $\forall x_1 \ldots x_n.\varphi$. $\square$

Corollary 1. *Assume k is the number of constants in a BSR theory $\mathcal{T}$, or 1 if $\mathcal{T}$ has no constant. If there is a $\mathcal{T}$-model of cardinality j, there is a finite $\mathcal{T}$-model with any cardinality i with $k \leq i \leq j$. If there is an infinite $\mathcal{T}$-model, there is a $\mathcal{T}$-model with any cardinality i with $k \leq i$.*

7.2 BSR theories and infinite models

We known that a BSR theory either has models for every finite and infinite cardinality greater than k, or it only has models of finite cardinalities all smaller than a number k'. What is missing is a way to decide if one theory has an infinite model or not. If it has no infinite model, the number k' can be computed (naïvely) by checking all finite models by increasing cardinalities until k' is found.

The following definition expresses some symmetry properties of models. We later show that the existence of an infinite model is equivalent to the existence of a finite model having such symmetry properties.

Definition 1. *Let $\mathcal{M}$ be an interpretation on domain D for a BSR theory $\mathcal{T}$. Let $A = \{\mathcal{M}[a] \,|\, a$ is a constant in $\mathcal{T}\}$ and $B = D \setminus A$. $\mathcal{M}$ is n-*repetitive *if $|B| \geq n$ and if there exists a total order $\prec$ on elements in B such that*

- *for every $m \leq n$;*
- *for every two strictly increasing (with respect to $\prec$) series $e_1, \ldots e_m$ and $e'_1, \ldots e'_m$ of elements in B;*
- *for every k-ary predicate symbol p used in $\mathcal{T}$;*
- *for every $d_1, \ldots d_k \in A \cup \{e_1, \ldots e_m\}$;*

$\mathcal{M}[p](d_1, \ldots d_k) = \mathcal{M}[p](d'_1, \ldots d'_k)$, *with* $d'_i = e'_j$ *if* $d_i = e_j$ *for some* j, $d'_i = d_i$ *otherwise. By extension, a theory is* n*-repetitive if it has a* n*-repetitive model.*

Observe that, thanks to Theorem 3, a theory is n-repetitive if it has a n-repetitive model $\mathcal{M}$ such that $|B| = n$, in the previous definition.

Example 5. Assume $\mathcal{T}$ is a theory with constants $a_1, \ldots a_{n_0}$, unary predicates $p^1_1, \ldots p^1_{n_1}$, binary predicates $p^2_1, \ldots p^2_{n_2}$.

$\mathcal{T}$ is 1-repetitive, if and only if $\mathcal{T} \cup R_1(b)$ is satisfiable, with

$$R_1(b) =_{\text{def}} \{b \neq a_1, \ldots b \neq a_{n_0}\}.$$

In other words, a theory $\mathcal{T}$ is 1-repetitive if it accepts a model with an element in the domain that is not assigned to a constant used in $\mathcal{T}$.

$\mathcal{T}$ is 2-repetitive, if and only if

$$\mathcal{T} \cup \bigcup_{i \in \{0,1\}} R_1(b_i) \ \cup \ R_2(b_0, b_1)$$

is satisfiable, with

$$\begin{aligned} R_2(b_0, b_1) =_{\text{def}} & \ \{b_0 \neq b_1\} \\ & \cup \{p^1_i(b_0) \equiv p^1_i(b_1) \mid i \in [1..n_1]\} \\ & \cup \{p^2_i(b_0, b_0) \equiv p^2_i(b_1, b_1) \mid i \in [1..n_2]\} \\ & \cup \{p^2_i(a_j, b_0) \equiv p^2_i(a_j, b_1) \mid i \in [1..n_2], \ j \in [1..n_0]\} \\ & \cup \{p^2_i(b_0, a_j) \equiv p^2_i(b_1, a_j) \mid i \in [1..n_2], \ j \in [1..n_0]\} \end{aligned}$$

$\mathcal{T}$ is 3-repetitive, if and only if

$$\mathcal{T} \cup \bigcup_{i \in \{0,1,2\}} R_1(b_i) \ \cup \bigcup_{\substack{i < j \\ i, j \in \{0,1,2\}}} R_2(b_i, b_j) \ \cup \ R_3(b_0, b_1, b_2)$$

is satisfiable, with

$$\begin{aligned} R_3(b_0, b_1, b_2) =_{\text{def}} & \ \{p^2_i(b_0, b_1) \equiv p^2_i(b_1, b_2) \equiv p^2_i(b_0, b_2) \mid i \in [1..n_2]\} \\ & \cup \{p^2_i(b_1, b_0) \equiv p^2_i(b_2, b_1) \equiv p^2_i(b_2, b_0) \mid i \in [1..n_2]\} \end{aligned}$$

Theorem 4. *If a BSR theory* $\mathcal{T}$ *with* n *distinct quantified variables has a* n*-repetitive model with cardinality* k*, then it has (*n*-repetitive) models with any (finite or infinite) cardinality* $k' \geq k$*.*

Proof. Assume $\mathcal{M}$ is a n-repetitive $\mathcal{T}$-model of cardinality k on domain D. Let A be $\{\mathcal{M}[a] \mid a \text{ is a constant in } \mathcal{T}\}$, and $B = D \setminus A$ ($|B| = k - |A| \geq n$). Assume also that $\prec$ is the total order on B mentioned in Definition 1. Choose a strictly increasing (with respect to $\prec$) series of n distinct elements $e_1, \ldots e_n \in B$.

Let E be such that $E \cap D = \emptyset$, and $|D \cup E| = k'$. We define an interpretation $\mathcal{M}'$ on domain $D' = D \cup E$. The total order $\prec$ on B is extended to $B \cup E$. We then require that $\mathcal{M}'[a] = \mathcal{M}[a]$ for every constant a in $\mathcal{T}$, and that, for every m-ary predicate p in $\mathcal{T}$ and every $d'_1, \ldots d'_m \in D'$:

– if $|\{d'_1, \ldots d'_m\} \setminus A| > n$, $\mathcal{M}'[p(d'_1, \ldots d'_m)]$ does not matter;
– if $\{d'_1, \ldots d'_m\} \subseteq D$, $\mathcal{M}'[p(d'_1, \ldots d'_m)] = \mathcal{M}[p(d'_1, \ldots d'_m)]$;
– otherwise, let $e'_1, \ldots e'_n$ be a strictly increasing series including all elements in $\{d'_1, \ldots d'_m\} \setminus A$. $\mathcal{M}'[p](d'_1, \ldots d'_k) = \mathcal{M}[p](d_1, \ldots d_k)$, with $d_i = e_j$ if $d'_i = e'_j$ for some j, $d_i = d'_i$ otherwise.

By construction, $\mathcal{M}'$ is n-repetitive.

Every formula in $\mathcal{T}$ is of the form $\forall x_1 \ldots x_m.\varphi(x_1, \ldots x_m)$, with $m \leq n$. For all elements $d'_1 \ldots d'_m \in D'$, the truth value for $\mathcal{M}'_{x_1/d'_1, \ldots x_m/d'_m}[\varphi(x_1, \ldots x_m)]$ is $\mathcal{M}_{x_1/d'_1, \ldots x_m/d'_m}[\varphi(x_1, \ldots x_m)]$ (i.e. true), if $\{d'_1, \ldots d'_m\} \subseteq D$. Otherwise, assume $e'_1, \ldots e'_n$ is a strictly increasing series including all elements in $\{d'_1, \ldots d'_m\} \setminus A$. Since the model $\mathcal{M}'$ is n-repetitive, then $\mathcal{M}'_{x_1/d'_1, \ldots x_m/d'_m}[\varphi(x_1, \ldots x_m)]$ is equal to $\mathcal{M}_{x_1/d_1, \ldots x_m/d_m}[\varphi(x_1, \ldots x_m)]$ (i.e. true) where $d_i = e_j$ if $d'_i = e'_j$ for some j, $d_i = d'_i$ otherwise. Finally, $\mathcal{M}'$ is a model of $\forall x_1 \ldots x_n.\varphi(x_1, \ldots x_m)$. □

Theorem 5. *If a BSR theory $\mathcal{T}$ has a model with a cardinality greater than a number k' computable from the theory, then it has a n-repetitive model on domain $D = A \cup B$, where $A = \{\mathcal{M}[a] \mid a \text{ is a constant in } \mathcal{T}\}$, $A \cap B = \emptyset$ and $|B| = n$.*

Proof. Assume $\mathcal{T}$ has a finite model $\mathcal{M}'$ on domain D'. We define the sets $A = \{\mathcal{M}'[a] \mid a \text{ is a constant in } \mathcal{T}\}$ and $B' = D' \setminus A$. Choose an order $\prec$ on B'. We now compute the size of B' so that there exists a n-repetitive model. A suitable k' can then be computed from $|B'|$.

Given two ordered (with respect to $\prec$) series $e_1, \ldots e_m$ and $e'_1, \ldots e'_m$ of elements in B', we will say that the configurations for $e_1, \ldots e_m$ and $e'_1, \ldots e'_m$ are the same if for every k-ary predicate p, and for every $d_1, \ldots d_k \in A \cup \{e_1, \ldots e_m\}$, $\mathcal{M}'[p](d_1, \ldots d_k) = \mathcal{M}'[p](d'_1, \ldots d'_k)$, with $d'_i = e'_j$ if $d_i = e_j$ for some j, $d'_i = d_i$ otherwise. Notice that there are only a finite number of different configurations for m elements in B': more precisely a configuration is made of at most $b = \sum_p [m + |A|]^{\text{arity}(p)}$ Boolean values, where the sum ranges on all predicates in the theory. Thus the number of different configurations is bounded by $C = 2^b$.

Understanding colors as being configurations, one can use Theorem 6 (in Appendix A) to state that, if $|B'| > f(n, N, C)$, then there exists a model of cardinality $|A| + N$ for $\mathcal{T}$ with the same configuration for any m ordered distinct elements. Recursively applying this procedure for every $m \in [1..n]$, it is possible to compute the cardinality k' of the original model so that there exists a n-repetitive model with the suitable cardinality. □

From both previous theorems:

Corollary 2. *Given a BSR theory $\mathcal{T}$ using n distinct quantified variables. $\mathcal{T}$ has an infinite model if and only if it has a n-repetitive model.*

Checking if a BSR theory $\mathcal{T}$ has an n-repetitive model is reduced to checking the satisfiability of another BSR theory $\mathcal{T}'$, basically, $\mathcal{T}$ augmented with some quantifier-free formulas. For formulas containing operators discussed in Section 3, we have $n \leq 3$, and predicates have an arity of at most 2: $\mathcal{T}'$ is given in Example 5. If $\mathcal{T}$ does not have an infinite model, then there is a maximum cardinality j for its models. The theory accepts a model for every cardinality between the number k of constants in $\mathcal{T}$ and j. This number j is bounded by a computable number k'. Unluckily, we currently lack an efficient (if there exists) way to compute this j. A naïve process to determine this number is to try every cardinality greater than k; the process will eventually terminate. Finally notice that this inefficient process is not necessary when combining a BSR theory with theories that only have infinite models.

8 Conclusions

In Section 3, we noticed that the use of some operators to encode sets, properties on relation,... would imply to have to verify the $\mathcal{T}$-satisfiability of FOL formulas with quantifiers. It was also shown that this satisfiability problem can be reduced to the satisfiability problem of literals in the combination of the theory $\mathcal{T}$ and another decidable theory, precisely a set of Bernays-Schönfinkel-Ramsey formulas.

Combining a BSR theory with the empty theory is possible, and this is the basis to build a decision procedure for formulas that contain uninterpreted functions and predicates, some operators on sets, relations,... A prototype has been built, and the first results are promising. When formulas containing operators from Section 3 have to be studied in some decidable (non-empty) theory $\mathcal{T}$, the combination process with the BSR theory is more complicated. The method presented in Section 6 is not in itself a practical procedure: its complexity prevents a direct application. However we believe that it can be the basis for a useful tool, with implementation-oriented improvements and proper heuristics.

We mainly target the B [2] and TLA+ [9] formal methods. Those language heavily rely on some set theories, and we believe that the results in this paper can help automating the proof of some parts of the verification conditions, which often mix arithmetic symbols, uninterpreted functions, and set operators. Verification conditions generated within those formal methods are usually small, within reach of a decision procedure even if it is inefficient. For verification conditions that are not fully within the language of the decision procedure, we built a certified (through proof reconstruction [8, 10]) cooperation between a proof assistant and the automated tool. At the present time, this cooperation can be used to delegate the proof of theorems from Isabelle to our prototype implementation (see Section 5), and have the proofs rechecked by the kernel of Isabelle, ensuring consistency of the whole cooperation of both tools.

A direction for further research is to investigate how to use the knowledge and engineering embedded in state-of-the-art first order provers (for instance [14, 13, 4]) to handle the BSR theories within a combination of decision procedures.

Acknowledgments: I am grateful to Yves Guiraud, Yuri Gurevich, Stephan Merz, Silvio Ranise, Christophe Ringeissen, and Duc-Khanh Tran for the interesting discussions on this subject.

References

1. Yices: An SMT solver. Available on http://yices.csl.sri.com/.
2. J.-R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, 1996.
3. M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 5, pages 273–333. Elsevier Science B.V., 2001.
4. P. Baumgartner, A. Fuchs, and C. Tinelli. Implementing the Model Evolution Calculus. In S. Schulz, G. Sutcliffe, and T. Tammet, editors, *Special Issue of the International Journal of Artificial Intelligence Tools (IJAIT)*, volume 15 of *International Journal of Artificial Intelligence Tools*, 2005.
5. H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, Inc., Orlando, Florida, 1972.
6. P. Fontaine. *Techniques for verification of concurrent systems with invariants*. PhD thesis, Institut Montefiore, Université de Liège, Belgium, Sept. 2004.
7. P. Fontaine and E. P. Gribomont. Combining non-stably infinite, non-first order theories. In W. Ahrendt, P. Baumgartner, H. de Nivelle, S. Ranise, and C. Tinelli, editors, *Selected Papers from the Workshops on Disproving and the Second International Workshop on Pragmatics of Decision Procedures (PDPAR 2004)*, volume 125 of *Electronic Notes in Theoretical Computer Science*, pages 37–51, July 2005.
8. P. Fontaine, J.-Y. Marion, S. Merz, L. P. Nieto, and A. Tiu. Expressiveness + automation + soundness: Towards combining SMT solvers and interactive proof assistants. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 3920 of *Lecture Notes in Computer Science*, pages 167–181. Springer-Verlag, 2006.
9. L. Lamport. *Specifying Systems*. Addison-Wesley, Boston, Mass., 2002.
10. S. McLaughlin, C. Barrett, and Y. Ge. Cooperating theorem provers: A case study combining HOL-light and CVC lite. *Electronic Notes in Theoretical Computer Science*, 144(2):43–51, 2006.
11. G. Nelson and D. C. Oppen. Simplifications by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems*, 1(2):245–257, Oct. 1979.
12. R. Nieuwenhuis and A. Oliveras. Decision Procedures for SAT, SAT Modulo Theories and Beyond. The BarcelogicTools. (Invited Paper). In G. Sutcliffe and A. Voronkov, editors, *12h International Conference on Logic for Programming, Artificial Intelligence and Reasoning, LPAR'05*, volume 3835 of *Lecture Notes in Computer Science*, pages 23–46. Springer, 2005.
13. A. Riazanov and A. Voronkov. The design and implementation of Vampire. *AI Communications*, 15(2):91–110, 2002.
14. S. Schulz. System Abstract: E 0.61. In R. Goré, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning (IJCAR)*, number 2083 in Lecture Notes in Artificial Intelligence, pages 370–375. Springer, 2001.
15. G. Sutcliffe and C. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.
16. C. Tinelli. Cooperation of background reasoners in theory reasoning by residue sharing. *Journal of Automated Reasoning*, 30(1):1–31, Jan. 2003.
17. C. Tinelli and C. G. Zarba. Combining non-stably infinite theories. In I. Dahn and L. Vigneron, editors, *First Order Theorem Proving*, volume 86.1 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

A n-monochromatic theorem

We define a *n-subset* of S to be a subset of n elements of S. An *n-overgraph* in S is a set of n-subsets of S. In particular, a 2-overgraph is a (undirected) graph. The *complete* n-overgraph of S is the set of all n-subsets of S, and its *size* is the cardinality of S. A n-overgraph G is *colored* with a set of colors C if there is a coloring function that assigns an element in C to every n-subset in G. In particular, a colored 2-overgraph (that is, a colored graph), is a graph where all edges are assigned a color. A colored n-overgraph is *monochromatic* if the coloring function assigns the same color to every n-subset. A colored n-overgraph A is a *sub-overgraph* of a colored n-overgraph B if each n-subset $S \in A$ belongs to B, and if the color associated to S is the same in both colored n-overgraphs.

Theorem 6. *There exists a computable function f such that, for every set of colors C, for every $n, N \in \mathbb{N}$, and every complete n-overgraph G colored with C, if the size of G is greater or equal to $f(n, N, |C|)$, there exists a complete monochromatic n-sub-overgraph of G of size greater or equal to N.*

Proof. We proceed by induction on n and the size of C.

Notice first that $f(n, N, 1) = N$ for every n, since a n-overgraph is colored with a unique color is monochromatic. Also, $f(1, N, 2) = 2N$, since a set of $2N$ elements that have one color in a pair $\{b, w\}$ contains at least N elements of the same color.

We now consider $f(n, N, 2)$. Assume G is a complete n-overgraph in S colored by c using colors in $\{b, w\}$. We build the series S_i and e_i such that

- $S_0 = S$
- e_i is any element in S_i
- To build S_{i+1}, we consider the complete $(n-1)$-overgraph in $S_i \setminus \{e_i\}$, colored by c_{e_i}, where c_{e_i} assigns to each $(n-1)$-subset A of $S_i \setminus \{e_i\}$ the color given by c to the n-subset $A \cup \{e_i\}$. Using the induction hypothesis, if $|S_i| \geq f(n-1, x, 2)$, there is a subset $S_{i+1} \subseteq S_i \setminus \{e_i\}$ such that $|S_{i+1}| \geq x$ and the complete $(n-1)$-overgraph of S_{i+1} colored by c_{e_i} is monochromatic.

Let B be the set of e_i such that the $(n-1)$-overgraph in S_{i+1} is colored by b, and W be the set of e_i such that the $(n-1)$-overgraph in S_{i+1} is colored by w. The n-overgraphs in B and W colored by c are monochromatic. To have $|B| \geq N$ or $|W| \geq N$ it is sufficient that $|S_{2N}| = n - 1$. Defining function g to be such that $g(*) = f(n-1, *, 2)$, it is sufficient to have $|S_0| \geq g^N(N)$

It remains to define $f(n, N, |C|)$ when $|C| > 2$. Assume G is a n-overgraph in S colored by c using colors in $C \cup k$ ($k \notin C$). We now consider all colors in C as one sole color; using the induction hypothesis, if $|S| \geq f(n, N', 2)$ then there exists $S' \subseteq S$ such that $|S'| \geq N'$ and the complete n-overgraph of S' colored by c only uses colors in C, or is monochromatic with color k. Any way, if one chooses N' as being greater than $f(n, N, |C|)$, there exists a subset S'' of S' such that $|S''| \geq N$ and the complete n-overgraph of S'' colored by c is monochromatic. We thus define $f(n, N, |C|+1) = f(n, f(n, N, |C|), 2)$. $\square$