BDI: A New Decidable First-order Clause Class

Manuel Lamotte-Schubert and Christoph Weidenbach

Max Planck Institute for Informatics, Campus E1 4,
D-66123 Saarbrücken, Germany
{lamotte,weidenbach}@mpi-inf.mpg.de

Abstract

$\mathcal{BDI}$ (Bounded Depth Increase) is a new decidable first-order clause class. It strictly includes known classes such as $\mathcal{PVD}$. The arity of function and predicate symbols as well as the shape of atoms is not restricted in $\mathcal{BDI}$. Instead the shape of "cycles" in resolution inferences is restricted such that the depth of generated clauses may increase but is still finitely bound. The BDI class is motivated by real world problems where function terms are used to represent record structures.

We show that the hyper-resolution calculus modulo redundancy elimination terminates on $\mathcal{BDI}$ clause sets. Employing this result to the ordered resolution calculus, we can also prove termination of ordered resolution on $\mathcal{BDI}$, yielding a more efficient decision procedure.

1 Introduction

Identifying decidable fragments of first-order logic has a long tradition in automated reasoning research. It started with the specification of decidable quantor prefix classes at the beginning of the 20th century (see [4] for an overview). After the invention of automated reasoning calculi, in particular resolution-based calculi, it moved to the identification of decidable clause classes (e.g., see [3, 15, 6, 9, 10, 1]) which then serve, e.g., as (background) fragments for effective reasoning on tree automata properties, reachability problems in security, knowledge representation formalisms, or data structures. Decidability is shown via termination of the resolution calculus on the clause class. In particular, the resolution calculus terminates on a set of clauses, if the length (number of literals in a clause) and depth (maximal depth of a literal in a clause) of newly generated clauses can be finitely bound. For all of the above mentioned clause classes the term depth of newly generated clauses by the respective resolution (superposition) strategy does not grow. To this end the depth and structure of terms is a priori restricted, e.g., in some of the above classes only shallow linear terms, like $f(x_1, \ldots, x_n)$, are permitted.

In this paper we define a new clause class called $\mathcal{BDI}$ (Bounded Depth Increase) where the term structure of clauses belonging to the class is not restricted at all. In addition, and in contrast to, e.g., the monadic classes, predicates may have an arbitrary number of arguments. An overall bounded term depth is guaranteed by restricting the form of recursive definitions for predicates that occur in the clause set. For the $\mathcal{BDI}$ class any considered resolvent with have depth at most $2n$ where n is the maximal depth of clause in the initial set (Theorem 4.2). By requiring that all variables occurring in a positive literal of a clause, also occur in a negative one of that clause, (positive) hyper-resolution generates only ground clauses (Lemma 4.1), implying together with the depth bound termination and therefore decidability of the $\mathcal{BDI}$ class.

For example, consider the $\mathcal{BDI}$ clause set

$$\begin{array}{lrcl}
(1) & & \rightarrow & P(f(a), h(a), a) \\
(2) & P(x, y, z) & \rightarrow & Q(x, y, f(g(x))), S(x, y) \\
(3) & Q(x, y, f(z)) & \rightarrow & R(f(g(x)), x, h(y)) \\
(4) & R(f(g(x)), y, h(z)) & \rightarrow & P(x, y, z) \\
(5) & P(a, b, c) & \rightarrow &
\end{array}$$

where clauses are written in implication form. The clauses (2)-(4) recursively define the predicate P. By resolving clauses (1) and (2) the clause

$$\rightarrow Q(f(a), h(a), f(g(f(a)))), S(f(a), h(a))$$

is generated causing an overall depth increase by the term $f(g(f(a)))$, the third argument of Q through the first argument of P. The deeper term is a result of x occurring at depth 0 in $P(x, y, z)$ in clause (2) and at depth 2 in the third argument of $Q(x, y, f(g(x)))$. In this case, we require that the third argument of Q cannot show up by resolving along the cycle (2)-(4) as a first argument of P. We ensure this by the concept of a watched argument (Definition 3.4). The terms at watched arguments of an atom are assumed to never increase during any derivation and argument positions holding terms with increased variable depth only depend on watched argument positions. For the example, the argument positions $1, 2$ of the predicates Q and P are watched and all atoms with predicates P, Q, satisfy this requirement (Definition 3.7-(iii)). A second increase in depth is potentially produced by clause (3), at the first argument of the R atom, where the clauses (2)-(4) also recursively define R. This clause does also not eventually generate terms of increasing depth, because all occurrences of R atoms in the clause set are similar (see Section 3), i.e., they have the same tree shape, and thus can only generate a bounded increase in depth (Definition 3.6). Finally, for the clauses (1), (4), (5) the depth of occurrences of variables in positive literals is smaller than their respective depth in negative literals. As a result, positive hyper-resolution terminates on the above clause set.

The $\mathcal{BDI}$ class is not included in any known decidable clause class. It obviously generalizes $\mathcal{PVD}$ [6]. The class of guarded formulae, originally proposed by Andréka et al. [12], was shown to be decidable through an effectively bounded finite model property. The first resolution decision procedure for the guarded fragment has been described by de Nivelle [13]. It has been further studied by Georgieva et al. [7] resulting in the fragment GF1$^-$, for which hyper-resolution is a decision procedure. The class GF1$^-$ includes function symbols but does not support non-guarded formulas. For example, a transitivity clause is not included in this fragment but contained in our class $\mathcal{BDI}$. Further classes that can be decided by resolution (superposition) without generating clauses with a term depth increase are the monadic class [3], the class of shallow sort theories [15], or classes related to tree automata [10].

Another related class is $\mathcal{BU}$ [8], which generalizes the set of all clause sets one can obtain from GF1$^-$, includes function symbols, and is also decidable by hyper-resolution. The class definition of $\mathcal{BU}$ takes special care of variables, for example, every non-positive functional clause must contain a covering negative literal which contains all the variables of the clause. Eventually this limits the depth of clauses generated by hyper-resolution. In $\mathcal{BDI}$, we don't require such conditions but instead limit the form of recursive definitions.

The class $\mathcal{BDI}$ is not artificial. It arose naturally out of our work on analyzing real-word authorization structures as they occur, e.g., in enterprise relationship systems [11]. When modeling business processes function terms occur out of newly created business objects. For example, a system user creating a purchase requisition out of a business requirement to buy some good x could result in clause like

$$\mathrm{User}(y), \mathrm{BusReq}(x), \mathrm{Authorized}(y, \mathrm{PurchReq}) \rightarrow \mathrm{ToBeReleased}(\mathrm{purchreq}(y, x))$$

that is depth increasing in x and y. Typically, business processes are not recursive. But proving properties of such processes and the underlying authorization setup requires in particular considering clauses of the form

$$\mathrm{ToBeReleased}(\mathrm{purchreq}(y, x)) \rightarrow \mathrm{Authorized}(y, \mathrm{PurchReq})$$

and now we have created a depth increasing, recursive clause structure. For this very simple example, it is obvious that the two clauses can not result in a non-terminating hyper-resolution derivation. However, for the real-world authorization set ups we have considered, the full power of our new $\mathcal{BDI}$ class is needed to guarantee termination.

The remaining of the paper is organized as follows. We first constitute the notational background including the ordered (hyper-)resolution calculus in Section 2. In Section 3, we formally define the new clause class $\mathcal{BDI}$. The main ideas of the termination proof for the hyper-resolution calculus applied to the new class are then presented in Section 4. Although hyper-resolution is a decision procedure for $\mathcal{BDI}$, in practice and in particular in the context of authorization proofs it is generally not a good choice, because it enumerates all ground consequences out of a given clause set. Therefore, in Section 5 we prove decidability of our classes for ordered resolution with selection. Given a concrete problem class, the ordered resolution calculus with selection offers a by far more fine grained control on the size of the eventually generated clause set. Finally, we conclude the paper by investigating the border between $\mathcal{BDI}$ and undecidable clause classes and end with some further discussion on related and future work.

2 Background

We follow the notation from [16] and [2]. We consider a first-order language without equality constructed over a signature $\Sigma = (\mathcal{F}, \mathcal{R})$, where $\mathcal{F}$ and $\mathcal{R}$ are non-empty, disjoint, in general infinite sets of function and predicate symbols, respectively. Every function or predicate symbol has some fixed arity. In addition, we assume a further, infinite set $\mathcal{X}$ of variable symbols disjoint from the symbols in Σ. Then the set of all *terms* $\mathcal{T}(\mathcal{F}, \mathcal{X})$ is defined as usual. A term not containing a variable is a *ground term.* If $t_1, \ldots, t_n$ are terms and $R \in \mathcal{R}$ is a predicate symbol with arity n, then $R(t_1, \ldots, t_n)$ is an *atom.* We sometimes write $R(\vec{t})$ as a shortened version of $R(t_1, \ldots, t_n)$ with arguments $t_1, \ldots, t_n$. An atom or the negation of an atom is called a *literal.* Disjunctions of literals are *clauses* where all variables are implicitly universally quantified. Clauses are often denoted by their respective multisets of literals where we write multisets in usual set notation. Alternatively to the multiset notation of clauses, we write clauses in implication form $\Gamma \rightarrow \Delta$ where the multiset Γ is called *antecedent* and the multiset Δ is called *succedent* of the clause. The atoms in Γ denote negative literals while the the atoms in Δ denote the positive literals in a clause.

A *position* is a word over the natural numbers. Let $f(t_1, \ldots, t_n)$ be a term. The set $\mathit{pos}(f(t_1, \ldots, t_n))$ of positions of a term is recursively defined as (i) the empty word ϵ is a position in any term t and $t|_\epsilon = t$ (ii) if $t|_p = f(s_1, \ldots, s_n)$, then $p.i$ is a position in t for all $i = 1, \ldots, n$ and $t|_{p.i} = s_i$. The term $t[p/s]$ is obtained from t by replacing $t|_p$ in t with s.

The function *vars* returns the set of variables for some term, atom, literal, clause. The *depth* of a term t is the maximal length of a position in the term: $\mathit{depth}(t) = \mathit{max}(\{\mathit{length}(p) \mid p \in \mathit{pos}(t)\}$. The *depth* of a literal $[\neg]P(t_1, \ldots, t_n)$ is the maximal depth of its terms: $\mathit{depth}([\neg]P(t_1, \ldots, t_n)) = \mathit{max}(\{\mathit{depth}(t_1), \ldots, \mathit{depth}(t_n)\})$. The depth of a clause is the maximal depth of its literals, and in the same manner, the depth of a set of literals is the maximal depth of its literals. Additionally, the function depth is extended to variables and clauses (or sequences of literals) $\mathit{depth}(x, C)$ returning the maximal depth of an occurrence of the variable $x \in \mathit{vars}(C)$ in a clause C.

Ordered resolution is defined with respect to a reduction ordering $\succ$ that is total on ground terms. A *reduction ordering* $\succ$ is a well-founded, transitive relation satisfying for all terms t, s, l, positions $p \in \mathit{pos}(l)$ and substitutions σ that whenever $s \succ t$ then $l[p/s\sigma] \succ l[p/t\sigma]$. Any (reduction) ordering $\succ$ on terms (atoms) can be extended to clauses by considering clauses as multisets of occurrences of atoms as described in [16].

For the termination proof of the $\mathcal{BDI}$ clause class, inferences are computed only using the below ordered hyper-resolution rule. All inferred clauses by hyper-resolution will be ground, so factoring actually becomes condensation. As usual the calculus is based on a reduction ordering $\succ$ that is total on ground terms. Inference rules add the clause(s) below the bar to the current clause set.

Definition 2.1 (Ordered Hyper-Resolution)**.** The inference

$$\frac{E_1, \ldots, E_n \rightarrow \Delta \qquad \rightarrow \Delta_i, E'_i \quad (1 \leq i \leq n)}{(\rightarrow \Delta, \Delta_1, \ldots, \Delta_n)\sigma}$$

where

(i) σ is the simultaneous *mgu* of $E_1, \ldots, E_n, E'_1, \ldots, E'_n$,

(ii) all $E'_i\sigma$ are strictly maximal in $(\rightarrow \Delta_i, E'_i)\sigma$

is called an *ordered hyper-resolution* inference.

Definition 2.2 (Ordered Resolution)**.** The inference

$$\frac{\Gamma_1 \rightarrow \Delta_1, E_1 \qquad E_2, \Gamma_2 \rightarrow \Delta_2}{(\Gamma_1, \Gamma_2 \rightarrow \Delta_1, \Delta_2)\sigma}$$

where

(i) σ is the *mgu* of E_1 and E_2,

(ii) no literal in Γ_1 is selected,

(iii) $E_1\sigma$ is strictly maximal in $(\Gamma_1 \rightarrow \Delta_1, E_1)\sigma$,

(iv) the atom $E_2\sigma$ is selected or it is maximal in $(E_2, \Gamma_2 \rightarrow \Delta_2)\sigma$, and no literal in Γ_2 is selected

is called an *ordered resolution* inference. If conditions (ii)-(iv) are dropped, the inference is called *resolution*.

Definition 2.3 (Factoring)**.** The inference

$$\frac{\Gamma \rightarrow \Delta, E_1, E_2}{(\Gamma \rightarrow \Delta, E_1)\sigma}$$

where

(i) σ is the *mgu* of E_1 and E_2,

(ii) no literal in Γ is selected,

(iii) $E_1\sigma$ is maximal in $(\Gamma \rightarrow \Delta, E_1, E_2)\sigma$

is called *factoring*.

For the purpose of this paper the reduction rules subsumption and condensation suffice. Nevertheless, the general superposition redundancy criterion is applicable. A clause $\Gamma_1 \rightarrow \Delta_1$ *subsumes* a clause $\Gamma_2 \rightarrow \Delta_2$ if for some substitution σ we have $\Gamma_1\sigma \subseteq \Gamma_2$ and $\Delta_1\sigma \subseteq \Delta_2$. A clause $\Gamma' \rightarrow \Delta'$ is a *condensation* of a clause $\Gamma \rightarrow \Delta$ if $\Gamma' \rightarrow \Delta'$ subsumes $\Gamma \rightarrow \Delta$ and $\Gamma' \rightarrow \Delta'$ is obtained from $\Gamma \rightarrow \Delta$ by instantiation and duplicate literal deletion.

Now the (ordered) hyper-resolution calculus consists of the rules (ordered) hyper-resolution, factoring, condensation, and subsumption deletion and the (ordered) resolution calculus consists of the rules (ordered) resolution, factoring, condensation, and subsumption deletion. We assume that reduction rules are applied exhaustively and before the application of any inference rule.

3 The Clause Class $\mathcal{BDI}$

The class $\mathcal{PVD}$ (positive variable dominated) [6] is the starting point for the class definition of $\mathcal{BDI}$. The class $\mathcal{PVD}$ has already been proven to be decidable by hyper-resolution in [5]. In contrast to the clause class $\mathcal{PVD}$ where the maximal depth of any derived clause by hyper-resolution does not exceed the maximal depth of its parent clauses, the class $\mathcal{BDI}$ permits to have such a growth of the term depth for a derived clause. Clearly, this relaxation requires additional restrictions in order to guarantee that hyper-resolution still remains a decision procedure for $\mathcal{BDI}$.

In the following, we start by defining some additional notions that are needed to define the class $\mathcal{BDI}$. Let p be an arbitrary position of a term s (atom, literal). We call p an *inner position* if there exists a position q in s such that $q = p.r$, $r \neq \epsilon$. Two atoms $P(t_1, \ldots, t_n)$ and $Q(s_1, \ldots, s_m)$ are called *similar* if $\mathit{pos}(P(t_1, \ldots, t_n)) = \mathit{pos}(Q(s_1, \ldots, s_m))$ and for all inner positions p we have $P(t_1, \ldots, t_n)|_p = Q(s_1, \ldots, s_m)|_p$, implying $P = Q$ and $n = m$.

Definition 3.1 ($\mathcal{PVD}$)**.** A clause $\Gamma \rightarrow \Delta$ is $\mathcal{PVD}$ [6] (Positive Variable Dominated) if

(i) $\mathit{vars}(\Delta) \subseteq \mathit{vars}(\Gamma)$ (Δ is ground for $\Gamma = \emptyset$),

(ii) $\mathit{depth}(x, \Delta) \leq \mathit{depth}(x, \Gamma)$ for all $x \in \mathit{vars}(\Delta)$.

Definition 3.2 (Depth Increasing)**.** We call a clause $C = \Gamma \rightarrow P(t_1, \ldots, t_n), \Delta$ *depth increasing* if there is a variable $x \in \mathit{vars}(C)$ and $\mathit{depth}(x, t_i) > \mathit{depth}(x, \Gamma)$ for some t_i where $1 \leq i \leq n$. The variable x is called a *depth increasing variable* in C, $P(t_1, \ldots, t_n)$ a *depth increasing atom* in C, P a *depth increasing predicate* in C, and i a *depth increasing argument position* of P.

We call a clause $C = \Gamma \rightarrow P(t_1, \ldots, t_n), \Delta$ *uniquely depth increasing* if C is depth increasing, and there is exactly one depth increasing argument position i of $P(t_1, \ldots, t_n)$ such that for all depth increasing variables $x \in \mathit{vars}(C)$ we have $\mathit{depth}(x, \{P(t_1, \ldots, t_{i-1}, x, t_{i+1}, \ldots, t_n), \Delta\}) \leq \mathit{depth}(x, \Gamma)$. Given a clause set N, we call a depth increasing clause $C \in N$ *uniquely depth increasing clause in N for the predicate P at argument position i* if there is no different depth increasing clause for the same predicate P in N with depth increasing argument position $j \neq i$.

In order to speak about recursive definitions of predicates or, alternatively, cycles between clauses, we need to establish a notion of reachability between predicate symbols of atoms occurring in (possibly different) clauses.

Definition 3.3 (Reachability)**.** Given a clause set N, a predicate Q *is reachable from P in one step* if there is a clause $(\Gamma, P(\vec{s}) \rightarrow Q(\vec{t}), \Delta) \in N$. A predicate R *is reachable from P* if the predicate R is reachable in one step from Q and Q is reachable from P, or if R is reachable from P in one step. Additionally, we say that R is *reachable from a depth increasing clause* $(\Gamma \rightarrow Q(\vec{t}), \Delta) \in N$, with depth increasing predicate Q, if R is reachable from Q.

Consider the following set N as a motivating example for the below definition of watched arguments:

$$\begin{array}{rrcl} (1) & & \rightarrow & P(f(a), b, c) \\ (2) & P(x, y, z) & \rightarrow & Q(f(x), y, z) \\ (3) & Q(x, y, z) & \rightarrow & P(x, y, z) \end{array}$$

(Hyper-)resolution on N computes infinitely many clauses of the form $Q(f^i(a), b, c)$. The reason is, in particular, the second clause, where the depth of the occurrence of x in the succedent (term $f(x)$) is strictly larger than its depth in the antecedent (term x). In order to exclude such a situation, we "watch" the non-increasing arguments of $Q(f(x), y, z)$, which are y, z (the second and third argument). Due to the existing cycle between the second and third clause, we also watch the second and third argument in the atoms with predicate symbol P. In the case of a depth increase comparing the maximal term depth of the atoms on the right hand side and the maximal term depth occurring in the atoms on the left hand side (as we have it for the second clause), we require for the second clause that only variables from the watched arguments occur inside the depth growing terms. This means for the example, that if only the variables y or z are arguments of the function f in the second clause, the infinite nesting does not occur.

Definition 3.4 (Watched arguments)**.** Let *warg* be a function from predicate symbols to sequences of direct argument positions such that if $\mathit{warg}(P) = [i_1, \dots, i_n]$ then $1 \leq i_j \leq m$, $0 \leq n \leq m$, and $i_j < i_k$ for $k < j$ where m is the arity of P. In case $\mathit{warg}(P) = [i_1, \dots, i_n]$ then any i_j is called a *watched argument* of P. The function *warg* is extended to atoms by:

$$\mathit{warg}(P(t_1, \dots, t_m)) = [P(t_1, \dots, t_m)|_{i_1}, \dots, P(t_1, \dots, t_m)|_{i_n}]\,.$$

Definition 3.5 (Origination)**.** Let N be a set of clauses. *Origination* is defined inductively by:

(i) For all input clauses $C \in N$ their literals $L \in C$ *originates* from C.

(ii) For all hyper-resolution derived clauses $\rightarrow \Delta\sigma, \Delta_1, \dots, \Delta_n$ from parent clauses $C = Q_1(s_{1,1}, \dots, s_{1,m_1}), \dots, Q_n(s_{n,1}, \dots, s_{n,m_n}) \rightarrow \Delta$, $D_i = \rightarrow Q_i(u_{i,1}, \dots, u_{i,m_i}), \Delta_i$, the literals $L\sigma \in \Delta\sigma$ *originate* from the clause C, and each $L'\sigma \in \Delta_i$ *originates* from the clause D_i.

Definition 3.6 (BDI-1)**.** Let N be a set of clauses and *warg* a watched argument function. A clause $C = \Gamma \rightarrow P_1(t_{1,1}, \dots, t_{1,n_1}), \dots, P_m(t_{m,1}, \dots, t_{m,n_m}), \Delta$ from N with $1 \leq i \leq m$ satisfies BDI-1 if C is depth increasing, and

(i) $\mathit{vars}(\{P_1(t_{1,1}, \dots, t_{1,n_1}), \dots, P_m(t_{m,1}, \dots, t_{m,n_m}), \Delta\}) \subseteq \mathit{vars}(\Gamma)$, and
$\mathit{depth}(x, \Delta) \leq \mathit{depth}(x, \Gamma)$ for all $x \in \mathit{vars}(\Delta)$

(ii) for all $C' = P_i(s_1, \dots, s_n), \Gamma' \rightarrow \Delta' \in N$ where $P_i(s_1, \dots, s_n)\sigma = P_i(t_{i,1}, \dots, t_{i,n_i})\sigma$ for some unifier σ, the atoms $P_i(s_1, \dots, s_n)$ and $P_i(t_{i,1}, \dots, t_{i,n_i})$ are similar, and for all depth increasing variables x, positions p, variables y, argument positions j where $t_{i,j}|_p = x$, $s_j|_p = y$ with $y \in (\mathit{vars}(P_i(s_1, \dots, s_n)) \cap \mathit{vars}(\Delta'))$ it holds $\mathit{depth}(y, \Delta') = 0$

(iii) for all $P_i(t_{i,1}, \dots, t_{i,n_i})$ holds $\mathit{warg}(P_i(t_{i,1}, \dots, t_{i,n_i})) = [\ ]$

(iv) for all atoms $Q_k(\vec{r}_k), R_l(\vec{v}_l) \in \Gamma$ where Q_k is reachable from a depth increasing clause in N and R_l is not reachable from a depth increasing clause holds

$$\mathit{vars}(P_i(t_{i,1}, \dots, t_{i,n_i})) \subseteq \bigcup_k \mathit{vars}(\mathit{warg}(Q_k(\vec{r}_k))) \cup \bigcup_l \mathit{vars}(R_l(\vec{v}_l))$$

(v) for all atoms $Q(\vec{r}) \in \Gamma$ holds

$$(\mathit{warg}(Q(\vec{r})) = [\ ]\text{or for all } R(\vec{v}) \in \Delta \text{ it holds } \mathit{warg}(Q(\vec{r})) = \mathit{warg}(R(\vec{v})))$$

BDI-1-(ii) ensures that any derived atom from a clause satisfying BDI-1 with increased depth (compared to its parent clauses) cannot further contribute to the growth in depth in the next hyper-resolution step where the atom with the increased depth is considered as a parent clause. The R atoms in the clause set in the introduction are an example.

BDI-1-(iv) prevents to have two consecutive depth increases in an argument when two consecutive hyper-resolution inference steps with depth increasing clauses take place. The following example assumes a hyper-resolution inference applied to a clause satisfying BDI-1 and the immediate previous hyper-resolution was performed on a clause satisfying BDI-2. Consider the following clause set N for this condition:

$$\begin{array}{lrcl} (1) & P(x,y) & \rightarrow & Q(f(g(y)),y) \\ (2) & & \rightarrow & P(a,f(a)) \\ (3) & Q(x,y) & \rightarrow & R(f(x),x) \\ (4) & R(f(x),x) & \rightarrow & P(f(x),x) \end{array}$$

Hyper-resolution between clause (1) satisfying BDI-2 and (2) yields a depth increased clause $\rightarrow Q(f(g(f(a))),f(a))$. Next, we can apply hyper-resolution on clause (3) satisfying BDI-1 and the previously derived clause. According to the construction of BDI-2 (see below), the first (increased) argument of the literal $Q(f(g(y)),y)$ in clause (1) is not watched, but the second argument is, i.e. $\mathit{warg}(Q(f(g(y)),y)) = [y]$. Consequently, the variable condition of BDI-1-(iv) permits only variables in $R(f(x),x)$ where $x \in \mathit{vars}(\mathit{warg}(Q(x,y))$ because Q is reachable from a depth increasing clause. However, the variable condition cannot be satisfied because the variable $x \notin \mathit{warg}(Q(f(g(y)),y))$.

BDI-1-(v) prevents position swapping of previously increased arguments in the non-depth increasing arguments of a clause satisfying BDI-1. The idea is to require for any atom in the succedent with a non-empty watched argument list that the watched argument list for all atoms in the antecedent is either identical or empty.

Definition 3.7 (BDI-2)**.** Let N be a set of clauses and *warg* a watched argument function. A clause $C = \Gamma \rightarrow P(t_1,\ldots,t_j,\ldots,t_n),\Delta$ from N satisfies BDI-2 if C is a uniquely depth increasing clause in N for the predicate P at argument position j, and

(i) $\mathit{vars}(\{P(t_1,\ldots,t_j,\ldots,t_n),\Delta\}) \subseteq \mathit{vars}(\Gamma)$

(ii) for all $i \neq j$ holds $t_j \notin \mathit{warg}(P(t_1,\ldots,t_n))$ and $t_i \in \mathit{warg}(P(t_1,\ldots,t_n))$

(iii) for all atoms $Q(s_1,\ldots,s_n) \in \Gamma$ where Q is reachable from P and $\mathit{vars}(Q(s_1,\ldots,s_n)) \cap \mathit{vars}(P(t_1,\ldots,t_n)) \neq \emptyset$:

  (1) $\mathit{arity}(Q) = \mathit{arity}(P)$

  (2) $\mathit{warg}(Q(s_1,\ldots,s_n)) = \mathit{warg}(P(t_1,\ldots,t_n))$

  (3) $\mathit{vars}(s_j) \cap \mathit{vars}(P(t_1,\ldots,t_n)) = \emptyset$

(iv) for all clauses $C' \in N$ with $C' = \Gamma' \rightarrow \Delta'$ which have an atom whose predicate is reachable from P, it holds for all atoms $Q(\vec{r}) \in \Gamma$ that

$$(\mathit{warg}(Q(\vec{r})) = [\ ]\text{or for all } R(\vec{v}) \in \Delta \text{ it holds } \mathit{warg}(Q(\vec{r})) = \mathit{warg}(R(\vec{v})))$$

(v) for all atoms $S(v_1,\ldots,v_m) \in \Delta$ and $Q_k(\vec{r}_k), R_l(\vec{v}_l) \in \Gamma$ where Q_k is reachable from a depth increasing clause and R_l is not reachable from a depth increasing clause holds

$$\mathit{vars}(S(v_1,\ldots,v_m)) \subseteq \bigcup_k \mathit{vars}(\mathit{warg}(Q_k(\vec{r}_k))) \cup \bigcup_l \mathit{vars}(R_l(\vec{v}_l))$$

Please note that condition BDI-2-(iii) implies that the depth increasing atom has at least two arguments. BDI-2-(iii) takes care of the depth inside the depth increasing atom of a clause satisfying BDI-2. In a clause set N with recursive predicate definitions, this condition restricts the way of increasing the depth in order to prohibit an unbounded growth of depth. BDI-2-(iv) prevents the “transfer” of a term with increased depth in a literal to another literal inside a different clause.

BDI-2-(iv) and BDI-2-(v) guarantee that depth increasing cycles cannot be used several times with the same depth increasing term, analogous to the corresponding conditions in BDI-1-(iv) and BDI-1-(v).

Consider the following set of clauses:

$$\begin{array}{lrcl}
(1) & P(x,y), Q(z,y) & \rightarrow & P(f(z),y) \\
(2) & P(x,y) & \rightarrow & Q(x,y) \\
(3) & & \rightarrow & P(a,b)
\end{array}$$

In this example, the clause (1) does not satisfy BDI-1, nor BDI-2, nor $\mathcal{PVD}$. It does not satisfy $\mathcal{PVD}$ because it is depth increasing, nor does it satisfy BDI-1 because the occurrence of the atom $P(f(z),y)$ is not similar to $P(x,y)$ occurring in clause (2) which is required by BDI-1-(ii). And eventually, it also does not satisfy the conditions of BDI-2, because there is the atom $Q(z,y)$, Q is reachable from P but BDI-2-(iii)-(3) is violated. The clauses (2) and (3) satisfy $\mathcal{PVD}$.

Definition 3.8 ($\mathcal{BDI}$)**.** Let N be a set of clauses and *warg* a watched argument function. The set N belongs to $\mathcal{BDI}$ (bounded depth increasing) if for all $C \in N$:

(i) C satisfies $\mathcal{PVD}$, or

(ii) C satisfies BDI-1, or

(iii) C satisfies BDI-2,

and, additionally, for two depth increasing clauses $\Gamma \rightarrow P(t_1, \ldots, t_n), \Delta$ and $\Gamma' \rightarrow Q(t'_1, \ldots, t'_{n'}), \Delta'$ with depth increasing predicates P and Q satisfying BDI-2

(iv) the predicate Q is not reachable from P and vice versa.

In the context of a clause set N satisfying $\mathcal{BDI}$, we can relax condition BDI-2-(iv) to apply only to clauses satisfying $\mathcal{PVD}$. Please note that we can have clauses in N which satisfy the conditions of both BDI-1 and BDI-2.

Consider the following set of clauses as an example to demonstrate and discuss the different syntactical conditions of the class $\mathcal{BDI}$:

$$\begin{array}{lrcl}
(1) & & \rightarrow & P(f(a),h(a),a) \\
(2) & P(x,y,z) & \rightarrow & Q(x,y,f(g(x))),\ S(x,y) \\
(3) & Q(x,y,f(z)) & \rightarrow & R(f(g(x)),x,h(y)) \\
(4) & R(f(g(x)),y,h(z)) & \rightarrow & P(x,y,z) \\
(5) & P(a,b,c) & \rightarrow & \\
(6) & P(x,y,z) & \rightarrow & T(y,z) \\
(7) & T(x,y) & \rightarrow & R(x,y,g(z))
\end{array}$$

A common requirement for all clauses is that the set of variables of the succedent of each clause is a subset of the set of variables of the antecedent of the same clause. Clause (7) violates this condition and is therefore not in $\mathcal{BDI}$. For the rest we only consider the clauses (1) to (6). The ground clauses (1) and (5) trivially satisfy $\mathcal{PVD}$, as well as clause (4). The clause (2) is depth increasing and satisfies BDI-2: The variables occurring in atoms different than $Q(x,y,f(g(x)))$ do not increase the term depth. Further, the predicate P of the atom $P(x,y,z)$ is reachable from Q through the clauses (2)-(3)-(4). P has the same arity than Q, the lists of watched arguments (i.e. all arguments except the depth increasing argument) can be defined identical, and the variable z does not occur inside the third argument of $Q(x,y,f(g(x)))$ (BDI-2-(iii)). Clause (3) satisfies BDI-1 because the occurrence of the atom $R(f(g(x)),x,h(y))$ in clause (3) is similar to the atom $R(f(g(x)),y,h(z))$ in clause (4) (BDI-1-(ii)). Furthermore, the variable x whose depth has been increased in clause (3) occurs with depth 0 in the atom

$P(x, y, z)$ in the succedent of clause (4). In addition, the atom in the succedent of clause (3) satisfies $\mathit{vars}(R(f(g(x)), x, h(y))) \subseteq \mathit{vars}(\mathit{warg}(Q(x, y, f(z))))$ (BDI-1-(iv)).

Note that checking membership of a clause set N in $\mathcal{BDI}$ can be done in time at most quadratic in the size of N. Membership in $\mathcal{PVD}$ can be checked in time linear in the size of N. While this test depth increasing atoms according to BDI-1 or BDI-2 can already be identified. In time at most quadratic in the size of N reachability between the predicates of those atoms and all other atoms can be established. Once reachability is established BDI-1 can be decided in linear time in the size of N. Note that for BDI-1 the watched arguments of the depth increasing predicate need to be set to the empty set (BDI-1-(iii)) and the watched arguments of reachable predicates have to be set accordingly (BDI-1-(iv) and BDI-1-(v)). Similarly, the BDI-2 conditions can be checked in linear time. All other argument positions except for the depth increasing one need to be watched and conditions BDI-2-(iv) and BDI-2-(v) can be established/checked in linear time once the reachable predicates are identified. Finally, condition Definition 3.8-(iv) is also linearly checked on the basis of an established reachability relation on the predicates.

4 Termination of Hyper-Resolution on $\mathcal{BDI}$

In order to decide $\mathcal{BDI}$, we use the hyper-resolution calculus. The aim is to show that any derivation from a given finite $\mathcal{BDI}$ clause set N terminates. It is well known that this is the case if the depth of terms in clauses as well as the number of different variables in clauses can be finitely bound. For the new class $\mathcal{BDI}$, hyper-resolution will only generate ground clauses, implying that for termination it is sufficient to provide an overall depth bound.

Lemma 4.1. Any clause derived by a hyper-resolution inference from an initial clause set N satisfying $\mathcal{BDI}$ is positive ground.

Proof. Follows from the variable condition $\mathit{vars}(\Delta) \subseteq \mathit{vars}(\Gamma)$ that holds for all clauses $\mathit{vars}(\Delta) \subseteq \mathit{vars}(\Gamma)$ satisfying $\mathcal{BDI}$. □

Because Factoring is applied only to positive clauses, and positive clauses derived by hyper-resolution inferences are always ground as stated in Lemma 4.1, the application of the factoring rule corresponds to condensation which amounts to the elimination of duplicate literals. So for $\mathcal{BDI}$ actually no factoring rule is needed for completeness. We still need to have a bound on the term depth of any derived clause.

Theorem 4.2. Let N be a finite set of $\mathcal{BDI}$ clauses and $d_N = 2 \cdot \max\{\mathit{depth}(\Delta) \mid \Gamma \rightarrow \Delta \in N\}$. Then the term depth of any clause C derived by the hyper-resolution calculus from N is smaller than d_N.

Proof (Outline). The proof is by induction of the length of a hyper-resolution derivation. The induction invariant implying the above statement is: for any clause C

(i) $\mathit{depth}(C) \leq d_N$

(ii) for all atoms $P(t_1, \dots, t_n) \in C$ with $\mathit{depth}(P(t_1, \dots, t_n)) > \frac{d_N}{2}$ holds:

  (iia) $\mathit{warg}(P(t_1, \dots, t_n)) \neq [\ ]$, $P(\vec{t})$ is reachable from a depth increasing clause satisfying BDI-2, and for all arguments $t_p \in \mathit{warg}(P(\vec{t}))$ holds that $\mathit{depth}(t_p) \leq \frac{d_N}{2}$, or

  (iib) $\mathit{warg}(P(t_1, \dots, t_n)) = [\,]$ and $P(t_1, \dots, t_n)$ originates (and A is therefore also reachable) from a depth increasing clause satisfying BDI-1.

□

5 From Hyper to Ordered Resolution

Hyper-resolution enumerates all ground facts from a given clause set. For many practical applications this is not feasible. For example, in the context of our authorization analysis, thousands of authorization definitions for a large number of users need to me modeled. They imply a huge number of derivable ground facts representing the exact authorization instantiations for all these users. Therefore, we want to employ a specific selection strategy on atoms in order to avoid the naive enumeration of all derivable positive ground clauses. Consider the following abstract, but real world, set of clauses as an example to sketch the idea. Assume 10000 ground atoms $\rightarrow A(a_i, b_j)$ relating authorizations a_i to possible values b_j. Assume 10000 ground atoms of the form $\rightarrow \mathit{Holds}(u_i, a_j)$ that assign authorizations a_j to users u_i. Then already a clause of the form $\mathit{Holds}(x, y), A(y, z) \rightarrow \mathit{Access}(x, z)$ results already in a potential quadratic (10k*10k) number of concrete access rights. However, in some business process, these rights are only needed in a very specific way, e.g., a clause of the form $P(x, y, z),\ \mathit{Access}(x_1, x),\ \mathit{Access}(x_1, y),\ \mathit{Access}(x_1, z) \ \rightarrow\ Q(x_1, y, z)$ requires three specific rights in order to derive $Q(x_1, y, z)$. If we can first select $P(x, y, z)$ in this clause, then the overhead of generating all access rights for all users in order to reason about $Q(x_1, y, z)$ can be prevented. Therefore, we want to turn ordered resolution with selection into a decision procedure for $\mathcal{BDI}$.

In general, ordered resolution is not a decision procedure for $\mathcal{BDI}$. However, as we will shoe below, the $\mathcal{BDI}$ class justifies two additional reduction rules that then make ordered resolution terminate.

Theorem 5.1. Let N be an unsatisfiable clause set of the class $\mathcal{BDI}$ and $d_N = 2 \cdot \max\{\mathit{depth}(\Delta) \mid (\Gamma \rightarrow \Delta) \in N\}$. Consider a hyper-resolution proof of the empty clause with ordering $\succ$. Then there is a (non-ground) ordered resolution proof of the empty clause with respect to $\succ$ and an arbitrary selection strategy such that $\mathit{depth}(C) \leq d_N$ for all clauses C derived in this ordered resolution proof.

Proof. By Theorem 4.2 there is a hyper-resolution proof of the empty clause where any generated clause does not exceed the depth bound d_N. Having a hyper-resolution proof for N with depth bound d_N, we can construct an inconsistent subset S of N and ground it by some constant such that all ground clauses still have depth bound d_N. By refutational completeness of the ordered resolution calculus with selection, we can derive the empty clause from S. Because all inferences are ground in the refutation of S, any derived ground clause respects the depth bound d_N. Using the standard lifting lemma, we can construct a non-ground refutation of the original set N where it still holds $\mathit{depth}(C) \leq d_N$ for all clauses C derived by the ordered resolution calculus with an arbitrary selection strategy. □

We exploit Theorem 5.1 by the following two paramterized reduction rules that eventually enable a finite saturation of a $\mathcal{BDI}$ clause set via ordered resolution with selection.

Definition 5.2 (Variable Condensation(k))**.** The reduction

$$\frac{C}{C\sigma_{1,2}, \ldots, C\sigma_{l-1,l}}$$

where $\mathit{vars}(C) = \{x_1, \ldots, x_l\}$, $l > k$ and $\sigma_{i,j} = \{x_i \mapsto x_j\}$ for all i, j with $1 \leq i < l$, $i < j \leq l$, is called *Variable Condensation.*

Definition 5.3 (Depth Cutoff(k))**.** The reduction

$$\frac{C}{}$$

where *depth*$(C) > k$ is called *Depth Cutoff.*

Theorem 5.4. Let N be a finite set of clauses satisfying $\mathcal{BDI}$ and Σ' be the signature symbols occurring in N. Then the ordered resolution calculus with an arbitrary selection strategy together with *Depth Cutoff* (d_N) and *Variable Condensation*(e_N) where $d_N = 2 \cdot \max\{\mathit{depth}(\Delta_C) \mid C \in N\}$ and $e_N = |\{t \mid t \in T(\Sigma'), \mathit{depth}(t) \leq d_N\}|$ is complete and terminating.

Proof. It follows from Theorem 5.1 that the standard ordered resolution calculus with an arbitrary selection strategy is able to derive the empty clause and none of the derived clauses in the ordered resolution proof exceeds the depth of d_N. Thus, if we have a clause D with *depth*$(D) > d_N$, we apply *Depth Cutoff* (d_N) on D and discard it as it will not be required to refute N in case of a contradiction.

Additionally, with respect to the finitely many signature symbols in N and the depth limit d_N only e_N many different ground terms need to be considered in any proof. Therefore, we can apply *Variable Condensation*(e_N) on any (derived) clause D such that the total number of different variables in any derived clause is bounded as well. □

6 Conclusion and Future Work

In order to emphasize the thin line between decidability and undecidability on our new class definition, we present two examples each violating only one of the conditions of $\mathcal{BDI}$, and show, that it is possible to encode the Post Correspondence Problem, PCP [14] using the relaxed conditions. For the PCP consider words over an alphabet $\{0, 1\}$. We construct a clause set such that an instance of the PCP problem has a solution if and only if the clause set is unsatisfiable. We encode words over '0', '1' by using terms built from the constant a and the monadic function symbols f_0, f_1. For example, the word 110 is represented as $f_1(f_1(f_0(a)))$. The corresponding string s for a term is denoted as $f_s(x)$. For a PCP instance $((u_1, v_1), (u_2, v_2), \ldots, (u_m, v_m))$, the overall clause set representing the PCP encoding is:

$$\begin{aligned} &\rightarrow \; P(f_{u_i}(a), f_{v_i}(a)) \quad 1 \leq i \leq m && (1.1)\\ P(x, y) &\rightarrow \; P(f_{u_i}(x), f_{v_i}(y)) \quad 1 \leq i \leq m && (1.2)\\ P(x, x) &\rightarrow && (1.3) \end{aligned}$$

The clauses of the form (1.1) represent the start state for m words and clauses (1.2) the recursion to construct larger words. Eventually, clause (1.3) neglects the existence of a common word.

Consider the below clause set of Example 6.1. The clauses (2.1) and (2.4) both satisfy condition $\mathcal{PVD}$ while the clauses (2.2) and (2.3) both satisfy BDI-2. In contrast to the standard formalization of the PCP problem, the extension of words (original clause (1.2)) is now spread over two clauses ((2.2) and (2.3)). However, these clauses in combination do not satisfy $\mathcal{BDI}$-(iv), because the predicate P is reachable from Q_i (and vice versa).

Example 6.1.

$$
\begin{array}{rcrl}
 & \rightarrow & P(f_{u_i}(a), f_{v_i}(a)) & 1 \leq i \leq m \qquad (2.1)\\
P(x,y) & \rightarrow & Q_i(f_{u_i}(x), y) & 1 \leq i \leq m \qquad (2.2)\\
Q_i(x,y) & \rightarrow & P(x, f_{v_i}(y)) & 1 \leq i \leq m \qquad (2.3)\\
P(x,x) & \rightarrow & & \qquad\qquad\qquad (2.4)
\end{array}
$$

So dropping the reachability condition of $\mathcal{BDI}$ leads to an undecidable clause class.

Consider the below clause set of Example 6.2. Here, we have used the same idea as in Example 6.1, namely, to distribute the extension of words over several clauses (3.2)-(3.5). The clauses (3.1) and (3.6) satisfy $\mathcal{PVD}$ while the remaining clauses are candidates to satisfy BDI-1. Starting from the clauses (3.2), the atoms with Q_i, R_i-predicates occurring in (3.3)-(3.5) are all similar, respectively. However, the variable condition of BDI-1-(iv) is violated in (3.3) and (3.4). Consider one of the clauses resulting from (3.3) as an example: The atom $Q_i(f_{u_i}(x), y)$ is reachable from a critical clause (a clause resulting from (3.2)), $\mathit{vars}(\mathit{warg}(Q_i(f_{u_i}(x), y))) = \emptyset$ and there are no other atoms that are not reachable from a critical clause on the left hand side. Consequently, the right hand side of the clause had to be ground to satisfy condition BDI-1-(iv).

Example 6.2.

$$
\begin{array}{rcrl}
 & \rightarrow & P(f_{u_i}(a), f_{v_i}(a)) & 1 \leq i \leq m \qquad (3.1)\\
P(x,y) & \rightarrow & Q_i(f_{u_i}(x), y), R_i(x, f_{v_i}(y)) & 1 \leq i \leq m \qquad (3.2)\\
Q_i(f_{u_i}(x), y) & \rightarrow & R_i(x, f_{v_i}(y)) & 1 \leq i \leq m \qquad (3.3)\\
R_i(x, f_{v_i}(y)) & \rightarrow & Q_i(f_{u_i}(x), y) & 1 \leq i \leq m \qquad (3.4)\\
Q_i(f_{u_i}(x), y), R_i(x, f_{v_i}(y)) & \rightarrow & P(f_{u_i}(x), f_{v_i}(y)) & 1 \leq i \leq m \qquad (3.5)\\
P(x,x) & \rightarrow & & \qquad\qquad\qquad (3.6)
\end{array}
$$

So dropping condition BDI-1-(iv) leads to an undecidable clause class.

In general, any violation of the conditions of BDI-1 or BDI-2 results in a clause class where hyper-resolution is no longer a decision procedure. The above two clause sets show that at least two of the conditions are mandatory in order to obtain a decidable clause class. As part of future work, we will investigate whether some conditions can be relaxed by appropriate refinements of the (hyper-)resolution calculus.

We have presented a new decidable clause class $\mathcal{BDI}$. It is motivated by our authorization analysis experiments. As hyper-resolution terminates on $\mathcal{BDI}$ it enjoys the finite model property. In addition, we showed that even any ordered resolution calculus with selection cutting off clauses with terms exceeding some a priori bound and variable condensing clauses exceeding a certain limit of different variables, decides the class. The this way extended ordered-resolution calculus can in fact efficiently decide properties for large $\mathcal{BDI}$ clause sets generated out of authorization structures.

There are ways to extend the $\mathcal{BDI}$ class or derive new decidable classes from it. An obvious modification would be to turn the variable conditions from succedent to antecedent and adopt the resolution strategy accordingly. Furthermore, it is possible to extend condition BDI-2 to several depth growing argument positions.

References

[1] Alessandro Armando, Maria Paola Bonacina, Silvio Ranise, and Stephan Schulz. New results on rewrite-based satisfiability procedures. *ACM Transactions on Computational Logic*, 10(1):4:1–4:51, 2009.

[2] Leo Bachmair and Harald Ganzinger. Resolution theorem proving. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 2, pages 19–99. Elsevier and MIT Press, 2001.

[3] Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Superposition with simplification as a decision procedure for the monadic class with equality. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *Computational Logic and Proof Theory, Third Kurt Gödel Colloquium*, volume 713 of *LNCS*, pages 83–96. Springer, August 1993.

[4] Egon Börger, Erich Grädel, and Yuri Gurevich. *The classical decision problem.* Perspectives in mathematical logic. Springer, 1996.

[5] C. Fermuller, T. Tammet, N. Zamov, and Alexander Leitsch. *Resolution Methods for the Decision Problem.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1993.

[6] Christian G. Fermüller, Alexander Leitsch, Ullrich Hustadt, and Tanel Tamet. Resolution decision procedures. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume II, chapter 25, pages 1791–1849. Elsevier, 2001.

[7] Lilia Georgieva, Ullrich Hustadt, and Renate A. Schmidt. Hyperresolution for guarded formulae. *J. Symbolic Computat*, 36:2003, 2000.

[8] Lilia Georgieva, Ullrich Hustadt, and RenateA. Schmidt. A new clausal class decidable by hyperresolution. In Andrei Voronkov, editor, *Automated DeductionCADE-18*, volume 2392 of *Lecture Notes in Computer Science*, pages 260–274. Springer Berlin Heidelberg, 2002.

[9] Ullrich Hustadt, Renate A. Schmidt, and Lilia Georgieva. A survey of decidable first-order fragments and description logics. *Journal of Relational Methods in Computer Science*, 1:251–276, 2004.

[10] Florent Jacquemard, Michaël Rusinowitch, and Laurent Vigneron. Tree automata with equality constraints modulo equational theories. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4130 of *Lecture Notes in Computer Science*, pages 557–571. Springer, 2006.

[11] Manuel Lamotte-Schubert and Christoph Weidenbach. Analysis of authorizations in SAP R/3. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *FTP 2009 : First-Order Theorem Proving*, volume 556 of *CEUR Workshop Proceedings*, pages 90–104, Oslo, Norway, July 2009. CEUR.

[12] Carsten Lutz, Ulrike Sattler, and Stephan Tobies. A suggestion for an n-ary description logic. In *Description Logics*, 1999.

[13] Hans De Nivelle. Resolution decides the guarded fragment., 1998. ILLC report CT-98-01, University of Amsterdam, The Netherlands.

[14] Emil L. Post. A variant of a recursively unsolvable problem. *J. Symbolic Logic*, 12(2):255–56, 1946.

[15] Christoph Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In Harald Ganzinger, editor, *16th International Conference on Automated Deduction, CADE-16*, volume 1632 of *LNAI*, pages 314–328. Springer, 1999.

[16] Christoph Weidenbach. Combining superposition, sorts and splitting. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 2, chapter 27, pages 1965–2012. Elsevier, 2001.