A Deductive System for FO(ID) Based on Least Fixpoint Logic ^{*}

Ping Hou and Marc Denecker

Department of Computer Science, Katholieke Universiteit Leuven, Belgium
{Ping.Hou, Marc.Denecker}@cs.kuleuven.be

Abstract. The logic FO(ID) uses ideas from the field of logic programming to extend first order logic with non-monotone inductive definitions. The goal of this paper is to extend Gentzen’s sequent calculus to obtain a deductive inference method for FO(ID). The main difficulty in building such a proof system is the representation and inference of unfounded sets. It turns out that we can represent unfounded sets by least fixpoint expressions borrowed from stratified least fixpoint logic (SLFP), which is a logic with a least fixpoint operator and characterizes the expressibility of stratified logic programs. Therefore, in this paper, we integrate least fixpoint expressions into FO(ID) and define the logic FO(ID,SLFP). We investigate a sequent calculus for FO(ID,SLFP), which extends the sequent calculus for SLFP with inference rules for the inductive definitions of FO(ID). We show that this proof system is sound with respect to a slightly restricted fragment of FO(ID) and complete for a more restricted fragment of FO(ID).

1 Introduction

Inductive definitions are common in mathematical practice. For instance, the non-monotone inductive definition of the satisfaction relation $\models$ can be found in most textbooks on first order logic (FO). This prevalence of inductive definitions indicates that these offer a natural and well-understood way of representing knowledge. It is well-known that, in general, inductive definitions cannot be expressed in first order logic.

It turns out, however, that certain knowledge representation logics do allow a natural and uniform formalization of the most common forms of inductive definitions. The authors of [6, 7] pointed out that semantical studies in the area of logic programming might contribute to a better understanding of such generalized forms of induction. In particular, it was argued that the well-founded semantics of logic programming [19] extends monotone induction and formalizes induction over well-founded sets and iterated induction. The language of FO(ID) uses the well-founded semantics to extend classical first order logic with a new “inductive definition” primitive. In the resulting formalism, all kinds of definitions regularly found in mathematical practice – e.g., monotone inductive

^{*} This work is supported by FWO-Vlaanderen and by GOA/2003/08.

definitions, non-monotone inductive definitions over a well-ordered set, and iterated inductive definitions – can be represented in a uniform way. Moreover, this representation neatly coincides with the form such definitions would take in a mathematical text. For instance, in FO(ID) the transitive closure of a graph can be defined as:

$$\left\{ \begin{array}{l} \forall x, y \text{ TransCl}(x, y) \leftarrow \text{Edge}(x, y) \\ \forall x, y \text{ TransCl}(x, y) \leftarrow (\exists z \text{ TransCl}(x, z) \wedge \text{TransCl}(z, y)) \end{array} \right\}.$$

However, FO(ID) is able to handle more than only mathematical concepts. Indeed, inductive definitions are also crucial in declarative Knowledge Representation. Not only non-inductive definitions are frequent in common-sense reasoning as argued in [2], also inductive definitions are. For instance, in [8], it was shown that situation calculus can be given a natural representation as an iterated inductive definition. The resulting theory is able to correctly handle tricky issues such as recursive ramifications, and is in fact, to the best of our knowledge, the most general representation of this calculus to date. It thus appears that FO(ID) has very strong links to several KR-paradigms.

As for every formal logical system, the development of deductive inference methods is an important research topic. For instance, it is well-known that deductive reasoning is a distinguished feature of Description Logics. However, because FO(ID) is not even semi-decidable, it is clear that a sound and complete proof system for FO(ID) cannot exist. As such, we will have to investigate deductive systems for FO(ID) and the subclasses of FO(ID) for which these systems are complete.

The initial motivation for the research of this paper is to extend Gentzen’s sequent calculus **LK** [12, 18] to obtain a proof system for FO(ID). We intended to build a sequent calculus for FO(ID) which is sound in general and complete for a useful subclass of FO(ID). The main challenge in building such a calculus is the representation and inference of unfounded sets [10]. In our approach to this problem, we represent unfounded sets by least fixpoint expressions borrowed from SLFP [5]. Compton described stratified least fixpoint logic (SLFP) in [5], which is a logic with a least fixpoint operator and characterizes the expressibility of stratified logic programs. He used sequent calculus to investigate deductive inference method for SLFP and proved the soundness and completeness of such a proof system. This work is an extension of the previous work in [14], which presents a sequent calculus for the propositional fragment of FO(ID), to the first order case.

In our deductive system, we use Compton’s inference system for SLFP. Our strategy consists of two steps. The first is to integrate least fixpoint expressions into FO(ID), leading to the logic FO(ID,SLFP). In the logic FO(ID,SLFP), we then build a proof calculus, which extends Compton’s sequent calculus for SLFP with inference rules for the inductive definitions of FO(ID). By restricting input to FO(ID), this proof calculus yields a proof system for FO(ID) then. However, in proofs of FO(ID) formulae, some inference rules may introduce least fixpoint expressions, and thus, proofs may contain expressions of FO(ID,SLFP).

The contributions of this paper can be summarized as follows: (a) We study a deductive inference method for FO(ID) based on stratified least fixpoint logic by defining the logic FO(ID,SLFP) and introducing a sequent calculus for FO(ID,SLFP). (b) We show that the deductive system is sound for a slightly restricted fragment of FO(ID,SLFP), where all definitions have to be total. Also, we investigate a more restricted fragment of FO(ID) and show the completeness result of the sequent calculus for FO(ID,SLFP) for this fragment.

This work is a step forward in the development of domain independent deductive reasoning for FO(ID). This form of inference is similar as that in Description Logics or classical logic but differs from Answer Set Programming which imposes Domain Closure. Our deductive calculus and the completeness result for a fragment of FO(ID), can be viewed as a step in developing tools similar as in the field of description logics. In a similar aim, a decidable guarded fragment of FO(ID) was described recently in [20].

The structure of this paper is as follows. We introduce FO(ID,SLFP) in Section 2. We present a deductive system for FO(ID,SLFP) in Section 3. The main results of the soundness and completeness of the deductive system are presented in Section 4. We finish with conclusions, related and future work.

2 Preliminaries

We start by defining the logic FO(ID,SLFP), which is the extension of FO with both inductive definitions and stratified least fixpoint expressions.

2.1 Syntax of FO(ID,SLFP)

In this subsection, we present the syntax of FO(ID,SLFP), which is an integration of FO(ID) and SLFP.

We assume familiarity with classical logic. A vocabulary Σ consists of Σ^c and Σ^v where Σ^c consists of (countable) predicate constants and function symbols while Σ^v is a (countable) set of predicate variables. Object symbols and propositional symbols are 0-ary function symbols, respectively predicate symbols. Terms and first order formulae (FO formulae) of Σ are defined as usual, and are built inductively from object symbols, function symbols, predicate symbols and logical connectives and quantifiers. A predicate symbol P has a negative (positive) occurrence in a formula F if P has an occurrence in the scope of an odd (even) number of occurrences of the negation symbol $\neg$.

First, we introduce the notation of a definition. A *definition* over Σ is a set of rules of the form $\forall \bar{x}(P(\bar{x}) \leftarrow \varphi)$, where $\bar{x}$ is a tuple of object variables over Σ^c , P is a predicate constant over Σ^c and φ is an FO formula over Σ^c such that all the free variables of φ occur in $\bar{x}$. We call $P(\bar{x})$ the *head* of the rule and φ the *body*. The connective $\leftarrow$ is called *definitional implication* and is to be distinguished from material implication $\supset$. A predicate appearing in the head of a rule of a definition D is called a *defined predicate* of D , any other symbol is called an *open symbol* of D . The set of defined predicates, respectively open

symbols of D are denoted by Σ_D^d , respectively, $\Sigma_D^o = \Sigma \setminus \Sigma_D^d$. Notice that there are no predicate variables occurring in the definition D .

Now we are ready to define the formulae of FO(ID,SLFP).

Definition 1. An FO(ID,SLFP) formula over Σ is defined by the following induction:

- If X is an n -ary predicate symbol (predicate constant or predicate variable) and $t_1, \dots, t_n$ are terms then $X(t_1, \dots, t_n)$ is a formula.
- If D is a definition then D is a formula.
- If ψ is a formula containing no free predicate variables, then $\neg\psi$ is a formula.
- If φ, ψ are formulae, then so are $\varphi \wedge \psi$ and $\varphi \vee \psi$.
- If ψ is a formula, then $\exists x\psi$ is a formula.
- If ψ is a formula containing no free predicate variables, then $\forall x\psi$ is a formula.
- If $\psi, \theta_1, \dots, \theta_n$ are formulae containing no definitions, $X_1, \dots, X_n$ are predicate variables, then $[LFP_{X_1(\bar{x}_1), \dots, X_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]\psi$ is a formula and called a stratified least fixpoint expression.

Note that the subformulae $\psi, \theta_1, \dots, \theta_n$ of a stratified least fixpoint expression $[LFP_{X_1(\bar{x}_1), \dots, X_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]\psi$ may not contain definitions, but may contain stratified least fixpoint expressions. Indeed, nesting of stratified least fixpoint expressions is allowed in FO(ID,SLFP), but nesting of definitions is not. All subformulae $\psi, \theta_1, \dots, \theta_n$ of an unnested stratified least fixpoint expression contain only positive occurrences of predicate variables.

Definition 2. Each occurrence of an object variable x in formulae $\exists x\varphi$ and $\forall x\varphi$ is bound. All occurrences of variables $\bar{x}_1, \dots, \bar{x}_n$ in the subformulae $\theta_1, \dots, \theta_n$ of a stratified least fixpoint expression $[LFP_{X_1(\bar{x}_1), \dots, X_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]\psi$ are bound. Each occurrence of a predicate variable X_i inside subformulae $\psi, \theta_1, \dots, \theta_n$ of a stratified least fixpoint expression $[LFP_{X_1(\bar{x}_1), \dots, X_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]\psi$ is bound. All other occurrences of an object or predicate variable is free. Let us denote $free(\varphi)$ the set of object and predicate variables with a free occurrence in φ .

An FO(ID,SLFP) sentence is a formula without free variables. An FO formula is an FO(ID,SLFP) formula without definitions and stratified least fixpoint expressions. An FO(ID) formula is an FO(ID,SLFP) formula without stratified least fixpoint expressions, and an SLFP formula is one without definitions. An FO(ID,SLFP) theory is a set of FO(ID,SLFP) sentences.

2.2 Semantics of FO(ID,SLFP)

The semantics of FO(ID,SLFP) is an integration of the semantics of FO(ID) and the semantics of SLFP.

The semantics of the FO(ID) is an integration of standard two-valued FO semantics with the well-founded semantics of definitions. For technical reasons, we introduce some concepts from three-valued logic. Consider the set of truth

values $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. The *truth order* $\leq$ on this set is induced by $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$; the *precision order* $\leq_p$ is induced by $\mathbf{u} \leq_p \mathbf{f}$ and $\mathbf{u} \leq_p \mathbf{t}$. Define $\mathbf{f}^{-1} = \mathbf{t}$, $\mathbf{u}^{-1} = \mathbf{u}$ and $\mathbf{t}^{-1} = \mathbf{f}$.

Given a domain $\mathcal{D}$, a *value* for an n -ary function symbol is a function from $\mathcal{D}^n$ to $\mathcal{D}$. A value for an n -ary predicate symbol is a function from $\mathcal{D}^n$ to $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$. A Σ -interpretation I consists of a domain $\mathcal{D}^I$, and a value σ^I for each symbol $\sigma \in \Sigma$. A two-valued interpretation is one in which predicates have range $\{\mathbf{t}, \mathbf{f}\}$. For each interpretation F for the function symbols of Σ , both truth and precision order have a pointwise extension to an order on all Σ -interpretations extending F . A domain atom of I is a tuple of a predicate $P \in \Sigma$ and a tuple $(a_1, \dots, a_n) \in \mathcal{D}^n$; it will be denoted by $P(a_1, \dots, a_n)$, or more compactly, $P(\bar{a})$.

For a given Σ -interpretation I , symbol σ and a value v for σ , we denote by $I[\sigma/v]$ the $\Sigma \cup \{\sigma\}$ -interpretation, that assigns to all symbols the same value as I , expect that $\sigma^{I[\sigma/v]} = v$. Likewise, for a domain atom $P(\bar{a})$ and a truth value $v \in \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, we define $I[P(\bar{a})/v]$ as the interpretation I' identical to I except that $P(\bar{a})^{I'} = P^{I'}(\bar{a}) = v$. Similarly, for any set U of domain atoms, $I[U/v]$ is identical to I except that all atoms in U have value v . When $\Sigma' \subseteq \Sigma$, we denote the restriction of a Σ -interpretation I to the symbols of Σ' by $I|_{\Sigma'}$.

When all symbols of term t are interpreted in I , we define its value t^I using the standard induction. The truth value φ^I of an FO sentence φ in I is defined by the standard induction on the subformula order.

We now define the semantics of definitions. Firstly, we generalize the well-known concept of an unfounded set [19].

Definition 3. *Given a definition D and a three-valued Σ -interpretation I , an unfounded set of D in I is a non-empty set U of defined domain atoms such that each $P(\bar{a}) \in U$ is unknown in I and for each rule $\forall \bar{x}(P(\bar{x}) \leftarrow \varphi(\bar{x})) \in D$, $\varphi(\bar{a})^{I[U/\mathbf{f}]} = \mathbf{f}$.*

The set of all unfounded sets of D in I is denoted by $\mathcal{U}_D(I)$. The *maximal unfounded set* of D in I is defined by $\bigcup_{X \in \mathcal{U}_D(I)} X$.

Definition 4. [10] *We define a well-founded induction of a definition D extending a Σ_D^o -interpretation I_O as a sequence $(I^\xi)_{\xi \leq \alpha}$ of three-valued Σ -interpretations extending I_O such that:*

- for every defined predicate symbol P , P^{I^0} is the constant function $\mathbf{u}$,
- for each limit ordinal $\lambda \leq \alpha$, $I^\lambda = \text{lub}_{\leq_p}(\{I^\xi \mid \xi < \lambda\})$, and
- for each ordinal ξ , $I^{\xi+1}$ relates to I^ξ in one of the following ways:
 1. $I^{\xi+1} := I^\xi[P(\bar{a})/\mathbf{t}]$, for some domain atom $P(\bar{a})$, unknown in I^ξ , such that for some rule $\forall \bar{x}(P(\bar{x}) \leftarrow \varphi(\bar{x})) \in D$, $\varphi(\bar{a})^{I^\xi} = \mathbf{t}$;
 2. $I^{\xi+1} := I^\xi[U/\mathbf{f}]$, where U is an unfounded set of D in I^ξ .

A well-founded induction is *terminal* if it can not be extended anymore.

It can be shown that each well-founded induction of D extending I_O is strictly increasing in precision and the limit of every terminal well-founded induction of D extending I_O is the *well-founded model* of D [10].

For a given three-valued interpretation I and definition D , we define $D^I = \mathbf{t}$ if I is the well-founded model of D extending $I|_{\Sigma_D^0}$, and $D^I = \mathbf{f}$ otherwise. We are now ready to define the semantics of FO(ID). An interpretation I satisfies an FO(ID) formula φ (is a model of φ) if I is two-valued and $\varphi^I = \mathbf{t}$. As usual, this is denoted by $I \models \varphi$. I satisfies an FO(ID) theory T if I satisfies every $\varphi \in T$. Note that the semantics is two-valued. The restriction to consider only two-valued well-founded models boils down to the requirement that a definition D should be *total*, i.e., should define the truth of all defined domain atoms (see [9]).

Definition 5. A definition D is *total* if for each Σ_D^0 -interpretation I_O , the well-founded model of D extending I_O is two-valued.

We now give the semantics of stratified least fixpoint expressions (see [5]). Let φ be the form of $[LFP_{X_1(\bar{x}_1), \dots, X_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]\psi$. Let α be an assignment to all variables in this expression other than $X_1, \dots, X_n, \bar{x}_1, \dots, \bar{x}_n$. A structure $\mathfrak{A}$ satisfies φ in the assignment α if $\mathfrak{A}$ satisfies ψ in α' , where α' is identical to α except it assigns the least relations to every X_i ($1 \leq i \leq n$) satisfying the implications $\forall \bar{x}_1 (X_1(\bar{x}_1) \supset \theta_1), \dots, \forall \bar{x}_n (X_n(\bar{x}_n) \supset \theta_n)$.

3 The deductive system for FO(ID,SLFP)

In this section, we present **LFO(ID,SLFP)**, a proof system for FO(ID,SLFP) based on Gentzen's sequent calculus **LK** for first order logic [12, 18].

First, we introduce some basic definitions and notations. Let capital Greek letters $\Gamma, \Delta, \dots$ denote (possibly empty) multisets of FO(ID,SLFP) formulae. Γ, Δ denotes $\Gamma \cup \Delta$ and Γ, φ denotes $\Gamma \cup \{\varphi\}$. By $\bigwedge \Gamma$, respectively $\bigvee \Gamma$, we denote the conjunction, respectively disjunction of all formulae in Γ . By $\neg \Gamma$, we denote the multiset obtained by taking the negation of each formula in Γ . By $\Gamma \setminus \Delta$, we denote the multiset obtained by deleting from Γ all occurrences of formulae that occur in Δ . Given an FO(ID,SLFP) formula φ , by $\varphi(x/t)$, we denote the formula obtained by substituting all free occurrences of the object variable x in φ by term t .

A *sequent* is an expression of the form $\Gamma \rightarrow \Delta$. Γ and Δ are respectively called the *antecedent* and *succedent* of the sequent and each formula in Γ and Δ is called a *sequent formula*. We will denote sequents by $S, S_1, \dots$. A sequent $\Gamma \rightarrow \Delta$ is *valid*, denote by $\models \Gamma \rightarrow \Delta$, if every model of $\bigwedge \Gamma$ satisfies some formula in Δ . The sequent $\Gamma \rightarrow$ is equivalent to $\Gamma \rightarrow \perp$ and $\rightarrow \Delta$ is equivalent to $\top \rightarrow \Delta$, where $\perp, \top$ are logical constants denoting *false* and *true*, respectively.

An *inference rule* is an expression of the form $\frac{S_1; \dots; S_n}{S}$ ($n \geq 0$). Each S_i is called a *premise* of the inference rule, S is called *consequence*. Intuitively, an inference rule means that S can be inferred, given that all $S_1, \dots, S_n$ are already inferred.

The sequent calculus for SLFP, mentioned in [5], contains an infinitary inference rule for stratified least fixpoint expression with countably infinite number of premises. In [5], it is shown that SLFP is not compact. It follows that we must

have some sort of infinitary rule in any complete sequent calculus for SLFP. Such infinitary rules are expressed in the form: $\frac{\Gamma_m \rightarrow \Delta_m \quad (m \in \omega)}{\Gamma \rightarrow \Delta}$.

The *initial sequents*, or *axioms*, of **LFO(ID,SLFP)** are the sequents of the form

$$\varphi, \Gamma \rightarrow \Delta, \varphi \quad \text{or} \quad \perp \rightarrow \Delta \quad \text{or} \quad \Gamma \rightarrow \top \quad \text{or} \quad \Gamma \rightarrow \Delta, t = t$$

The inference rules for **LFO(ID,SLFP)** consists of the structural rules, logical rules, equality rules, the stratified least fixpoint rules and definition rules. We use the standard structural rules, logical rules and equality rules as given in many sources (see e.g. [4]).

The inference rules for the stratified least fixpoint expressions follow those in [5], which are of the form:

Stratified least fixpoint rules

$$\begin{aligned} \text{left: } & \frac{\Gamma, [LFP_{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]_m \psi \rightarrow \Delta \quad (m \in \omega)}{\Gamma, [LFP_{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)] \psi \rightarrow \Delta}; \\ \text{right: } & \frac{\Gamma \rightarrow \Delta, [LFP_{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)]_m \psi}{\Gamma \rightarrow \Delta, [LFP_{P_1(\bar{x}_1), \dots, P_n(\bar{x}_n)}(\theta_1, \dots, \theta_n)] \psi} \end{aligned}$$

Notice that the left stratified least fixpoint rule is infinitary: it has countably many premises while in the right stratified least fixpoint rule there is just one premise: m is a fixed nonnegative integer.

To these rules we add the *definition rules* of **LFO(ID,SLFP)**. The definition rules consist of the *right definition rule*, the *left definition rule* and the *definition introduction rule*. The left definition rule, respectively, the right definition rule, can introduce the defined predicates in the antecedents, respectively, in the succedents, of the sequents. The definition introduction rule can introduce the total definitions in the succedents of the sequents. Without loss of generality, we assume from now on that there is only one rule with the head $P(\bar{x})$ in a definition D for every $P \in \Sigma_D^d$. We refer to this rule as *the rule for P in D* and denote it by $\forall \bar{x}(P(\bar{x}) \leftarrow \varphi_P(\bar{x}))$. Indeed, any set of rules $\{\forall \bar{x}(P(\bar{x}) \leftarrow \varphi_1), \dots, \forall \bar{x}(P(\bar{x}) \leftarrow \varphi_n)\}$ can be transformed into a single rule $\forall \bar{x}(P(\bar{x}) \leftarrow \varphi_1 \vee \dots \vee \varphi_n)$.

The right definition rule allows inferring the truth of a defined atomic formula from a definition D . It is closely related to the step (1) of Definition 4. Let D be a definition and P a defined predicate of D . Then the right definition rule for P is given as follows.

$$\text{Right definition rule for } P \quad \frac{\Gamma \rightarrow \Delta, \varphi_P(\bar{t})}{D, \Gamma \rightarrow \Delta, P(\bar{t})}.$$

Example 1. Consider the definition D of the transitive closure of a directed graph G as $\{\forall x, y(T(x, y) \leftarrow G(x, y) \vee \exists z(T(x, z) \wedge T(z, y)))\}$, $\Gamma = \{G(a, b), G(b, c)\}$ and Δ an empty set. The instance of the right definition rule for $T(a, c)$ is

$$\frac{G(a, b), G(b, c) \rightarrow G(a, c) \vee \exists z(T(a, z) \wedge T(z, c))}{D, G(a, b), G(b, c) \rightarrow T(a, c)}.$$

The left definition rule allows inferring the falsity of defined atoms from a definition D and is therefore related to step (2) of Definition 4. We first introduce some notations. Given a set U of predicate symbols, let U' be a set consisting of new predicate symbol P' for every $P \in U$. The vocabulary Σ augmented with these new symbols is denoted by Σ' . Given an FO formula φ , φ' denotes the formula obtained by replacing all positive occurrences of every $P \in U$ in φ by P' . We call φ' the *renaming* of φ with respect to U . By $\neg\varphi'$, we mean $\neg(\varphi')$.

Let D be a definition and $U = \{P_1, \dots, P_n\}$ a non-empty set of defined predicates of D . Let $\psi_1(\bar{x}^1), \dots, \psi_n(\bar{x}^n)$ be n arbitrary FO formulae over Σ or stratified least fixpoint expressions over Σ with $\bar{x}^i$ as many fresh variables as the arity of $P_i(\bar{x}^i)$. Then the left definition rule for $(\{P_1(\bar{x}^1) \mid \psi_1(\bar{x}^1)\}, \dots, \{P_n(\bar{x}^n) \mid \psi_n(\bar{x}^n)\})$ is given as follows.

Left definition rule for $\{P_i(\bar{x}^i) \mid \psi_i(\bar{x}^i)\}_{i \in [1, n]}$

$$\frac{\begin{array}{c} \Gamma, \forall \bar{x}^1 (\psi_1(\bar{x}^1) \supset \neg P'_1(\bar{x}^1)), \dots, \forall \bar{x}^n (\psi_n(\bar{x}^n) \supset \neg P'_n(\bar{x}^n)) \rightarrow \Delta, \forall \bar{x}^1 (\psi_1(\bar{x}^1) \supset \neg \varphi'_{P_1}(\bar{x}^1)) \\ \vdots \\ \Gamma, \forall \bar{x}^1 (\psi_1(\bar{x}^1) \supset \neg P'_1(\bar{x}^1)), \dots, \forall \bar{x}^n (\psi_n(\bar{x}^n) \supset \neg P'_n(\bar{x}^n)) \rightarrow \Delta, \forall \bar{x}^n (\psi_n(\bar{x}^n) \supset \neg \varphi'_{P_n}(\bar{x}^n)) \end{array}}{D, \Gamma, \exists \bar{x}^i (\psi_i(\bar{x}^i) \wedge P_i(\bar{x}^i)) \rightarrow \Delta}$$

where Γ and Δ are multisets of FO(ID,SLFP) formulae over Σ and $\varphi'_{P_i}(\bar{x}^i)$ is the renaming of $\varphi_{P_i}(\bar{x}^i)$ with respect to U .

Actually, in the left definition rule, the set of atoms $\{P_1(\bar{x}^1) \mid \psi_1(\bar{x}^1)\} \cup \dots \cup \{P_n(\bar{x}^n) \mid \psi_n(\bar{x}^n)\}$ is a symbolic representation of a candidate unfounded set of D .

Example 2. Given the definition D of the transitive closure of a directed graph G as $\{\forall x, y (T(x, y) \leftarrow G(x, y) \vee \exists z (T(x, z) \wedge T(z, y)))\}$, $\Gamma = \{\neg \exists x G(x, a)\}$ and Δ an empty set. The instance of the left definition rule for $(\{T(x, y) \mid y = a\})$ is as follows:

$$\frac{\begin{array}{c} \neg \exists x G(x, a), \forall x, y (y = a \supset \neg T'(x, y)) \rightarrow \\ \forall x, y (y = a \supset \neg G(x, y) \wedge \neg \exists z (T'(x, z) \wedge T'(z, y))) \end{array}}{D, \neg \exists x G(x, a), \exists x, y (y = a \wedge T(x, y)) \rightarrow}$$

Notice that the set $\{T(x, y) \mid y = a\}$ is the symbolic representation of a unfounded set of D here.

The definition introduction rule allows inferring the truth of a total definition from FO(ID,SLFP) formulae. We introduce some notations. Let D be a total definition. Denote by P^R a new defined predicate for each $P \in \Sigma_D^d$. Denote by Σ^R the vocabulary $\Sigma \cup \{P^R \mid P \in \Sigma_D^d\}$. Denote by D^R the definition over the new vocabulary Σ^R obtained by replacing each occurrence of each defined predicate P in D by P^R . Let Γ and Δ be multisets of FO(ID,SLFP) formulae over Σ . The definition introduction rule for D is given as follows.

Definition introduction rule

$$\frac{\Gamma, D^R \rightarrow \forall \bar{x}^1 (P_1^R(\bar{x}^1) \equiv P_1(\bar{x}^1)), \Delta; \dots; \Gamma, D^R \rightarrow \forall \bar{x}^n (P_n^R(\bar{x}^n) \equiv P_n(\bar{x}^n)), \Delta}{\Gamma \rightarrow \Delta, D},$$

where $P_1, \dots, P_n$ are all defined predicates of D .

Example 3. Given a definition $D = \{ \forall x (P(x) \leftarrow P(x) \wedge O(x)) \}$, $\Gamma = \{ \forall x O(x), \forall x \neg P(x) \}$ and Δ an empty set. The instance of the definition introduction rule for D is as follows:

$$\frac{D^R, \forall x O(x), \forall \neg P(x) \rightarrow \forall x (P^R(x) \equiv P(x))}{\forall x O(x), \forall x \neg P(x) \rightarrow D},$$

where $D^R = \{ \forall x (P^R(x) \leftarrow P^R(x) \wedge O(x)) \}$.

It is necessary to emphasize that the definition D in the definition introduction rule is required to be total. We will give an example to show that this inference rule is not sound if D is not total. Actually, there are practically important syntactic classes of definitions which are known to be total, including positive definitions, stratified definitions and definitions over a well-founded order (see [9]).

We now come to the notion of an **LFO(ID,SLFP)**-proof for a sequent.

Definition 6. A proof in **LFO(ID,SLFP)** or **LFO(ID,SLFP)**-proof for a sequent S , is a tree T of sequents with root S . Moreover, each leaf of T must be an axiom and for each interior node S' there exists an inference rule such that S' is the consequence of that inference rule while the children of S' are precisely the premises of that inference rule. T is called a proof tree for S . A sequent S is called provable in **LFO(ID,SLFP)**, or **LFO(ID,SLFP)**-provable, if there is an **LFO(ID,SLFP)**-proof for it.

4 Main results

In this section, we will present that the deductive system **LFO(ID,SLFP)** is sound for a slightly restricted fragment of **FO(ID,SLFP)** and it is complete with respect to a more restricted fragment of **FO(ID,SLFP)**.

It is trivial to verify that all axioms of **LFO(ID,SLFP)** are valid and that the structural and logical rules and the rules for equality are sound. The soundness of the stratified least fixpoint rules can be shown analogously to that in [5]. Hence, only the soundness of the definition rules of **LFO(ID,SLFP)** must be presented.

Lemma 1 (Soundness of the right definition rule). If $\models \Gamma \rightarrow \Delta, \varphi_P(\bar{t})$ then $\models D, \Gamma \rightarrow \Delta, P(\bar{t})$.

Lemma 2 (Soundness of the left definition rule). If $\models \Gamma, \forall \bar{x}^1 (\psi(\bar{x}^1) \supset \neg P'_1(\bar{x}^1)), \dots, \forall \bar{x}^n (\psi_n(\bar{x}^n) \supset \neg P'_n(\bar{x}^n)) \rightarrow \Delta, \forall \bar{x}^i (\psi_i(\bar{x}^i) \supset \neg \varphi'_{P_i}(\bar{x}^i))$ for every $i \in [1, n]$, then $\models D, \Gamma, \exists \bar{x}^i (\psi_i(\bar{x}^i) \wedge P_i(\bar{x}^i)) \rightarrow \Delta$ for all $i \in [1, n]$.

Lemma 3 (Soundness of the definition introduction rule). *Let D be a total definition. If $\models D^R, \Gamma \rightarrow \Delta, \forall \bar{x}(P(\bar{x}) \equiv P^R(\bar{x}))$ for every $P \in \Sigma_D^d$, then $\models \Gamma \rightarrow \Delta, D$.*

Note that the definition introduction rule is not sound if D is not total. We illustrate it with an example.

Example 4. Given a definition $D = \{ \forall x(P(x) \leftarrow \neg P(x)) \}$ and Γ an empty set. It is obvious that D^R is non-total. Thus, $\models D^R \rightarrow \forall x(P^R(x) \equiv P(x))$ but $\not\models \rightarrow D$, which shows that the definition introduction rule is not sound when D is non-total.

By the fact that all inference rules in **LFO(ID,SLFP)** are sound if all definitions occurring in them are total and a straightforward induction, the soundness of **LFO(ID,SLFP)** can now be obtained.

Theorem 1 (Soundness of LFO(ID,SLFP)). *If a sequent of $\Gamma \rightarrow \Delta$ is provable in **LFO(ID,SLFP)** and all definitions in Γ and Δ are total, then $\models \Gamma \rightarrow \Delta$.*

In the following, we present the completeness property of **LFO(ID,SLFP)** for a fragment of FO(ID,SLFP). First, we define a special class of definitions.

Definition 7. *A definition D is an $\mathcal{S}$ -definition if the body of each rule in D satisfies the following conditions:*

- *it is in negation normal form (NNF).*
- *all occurrences of defined predicates in its subformula $\exists x\varphi$ are positive.*
- *all occurrences of defined predicates in its subformula $\forall x\varphi$ are negative.*
- *all occurrences of open predicates are in arbitrary ways.*

Example 5. $D = \{ \forall x, y(T(x, y) \leftarrow G(x, y) \vee \exists z(T(x, z) \wedge T(z, y))) \}$ is an $\mathcal{S}$ -definition. $D = \left\{ \begin{array}{l} \forall x, y(T(x, y) \leftarrow G(x, y) \vee \exists z(Q(x, z) \wedge \neg Q(z, y))) \\ \forall x, y(T(x, y) \leftarrow \neg O(x, y) \vee \forall z(\neg T(x, z) \vee T(z, y))) \end{array} \right\}$ is not an $\mathcal{S}$ -definition because there is a positive occurrence of T in $\forall z(\neg T(x, z) \vee T(z, y))$ while there is a negative occurrence of Q in $\exists z(Q(x, z) \wedge \neg Q(z, y))$.

It is obvious that some familiar types of definitions such as non-recursive definitions and positive definitions [9] are $\mathcal{S}$ -definitions. However, not all $\mathcal{S}$ -definitions are total. For example, $\forall x(P(x) \leftarrow \neg P(x))$ is an $\mathcal{S}$ -definition but it is not total.

An *SFO(ID,SLFP) formula* is an FO(ID,SLFP) formula with the restriction that all definitions occurring in it are $\mathcal{S}$ -definitions. **SFO(ID,SLFP)** is the complete fragment of FO(ID,SLFP) with respect to **LFO(ID,SLFP)**, as shown later.

The maximal unfounded set of an $\mathcal{S}$ -definition can be represented by the negation of stratified least fixpoint expressions, which is an important property of $\mathcal{S}$ -definitions and will be applied in the completeness proof. It is demonstrated by the following proposition, where the notation of φ'_{P_i} is same as that in the left definition rule and $\varphi'_{P_i}(P'_1/X_1, \dots, P'_m/X_m)$ denotes the formula obtained by replacing every occurrence of every P'_j ($j \in [1, m]$) in φ'_{P_i} by X_j .

Proposition 1. *Let D be an $\mathcal{S}$ -definition and $(I^\xi)_{\xi \leq \alpha}$ a well-founded induction of D . Let $I^{\xi+1} = I^\xi[U/\mathbf{f}]$ where U is the maximal unfounded set of D in I^ξ . Suppose that $\{P_1, \dots, P_m\}$ is the set of all defined predicates occurring in U . Then $\{P_1(x^1) \mid \neg([LFP_{X_1(x^1), \dots, X_m(x^m)}](\varphi'_{P_1}(P'_1/X_1, \dots, P'_m/X_m), \dots, \varphi'_{P_m}(P'_1/X_1, \dots, P'_m/X_m))][X_1(x^1)])\} \cup \dots \cup \{P_m(x^1) \mid \neg([LFP_{X_1(x^1), \dots, X_m(x^m)}](\varphi'_{P_1}(P'_1/X_1, \dots, P'_m/X_m), \dots, \varphi'_{P_m}(P'_1/X_1, \dots, P'_m/X_m))][X_m(x^m)])\}$ is a symbolic representation of U .*

We now turn to show the completeness of **LFO(ID,SLFP)** for **SFO(ID,SLFP)**. The proof is an extension of the direct style of completeness proof for Gentzen's **LK** as given in e.g. [4]. The structure of the completeness proof is then roughly as follows:

1. Assume that $\Gamma \rightarrow \Delta$ is not provable, construct from $\Gamma \rightarrow \Delta$ a *limit sequent* $\Gamma_\omega \rightarrow \Delta_\omega$, where Γ_ω and Δ_ω are infinite sets, such that no finite subsequent of $\Gamma_\omega \rightarrow \Delta_\omega$ is provable by using a uniform proof-search procedure, namely *schedule*, on which every formula of **FO(ID,SLFP)** appears infinitely often.
2. Define an equivalence relation $\sim$ on the terms of Σ that essentially factors out the equality formulae appearing in Γ_ω and use $\Gamma_\omega \rightarrow \Delta_\omega$ to construct a partial Σ -interpretation I_ω that interprets all symbols in Σ except all defined predicates.
3. Prove that $\Gamma_\omega \setminus \Pi_\omega \rightarrow \Delta_\omega \setminus \Upsilon_\omega$ is false in I_ω , where Π_ω is the multiset of all definitions appearing in Γ_ω and Υ_ω is the multiset of all definitions appearing in Δ_ω .
4. Use I_ω to construct a model M_ω and prove that M_ω is the well-founded model for every definition appearing in Γ_ω and M_ω is not the well-founded model for every definition appearing in Δ_ω .
5. It now follows from step 3 and step 4 that every finite subsequent of $\Gamma_\omega \rightarrow \Delta_\omega$ is false in M_ω , including $\Gamma \rightarrow \Delta$, so $\Gamma \rightarrow \Delta$ is not valid.

Steps 1,2,3 and the analogous version of step 5 also appears in standard completeness proof for first order logic. However, in our proof, the construction of the limit sequent in step 1 must take into account of the stratified least fixpoint rules. Applications of stratified least fixpoint rules must also be accounted for in step 3. The new work in our proof goes into establishing that M_ω defined in step 4 is indeed a well-founded model for every definition appearing in Γ_ω but M_ω is not a well-founded model for every definition appearing in Δ_ω , where the Proposition 1 is applied.

Theorem 2 (Completeness of SFO(ID,SLFP)). *Let Γ, Δ be multisets of **SFO(ID,SLFP)** formulae. If $\Gamma \rightarrow \Delta$ is valid and all definitions occurring in Γ and Δ are total, then $\Gamma \rightarrow \Delta$ is provable in **LFO(ID,SLFP)**.*

5 Conclusions, related and future work

We present a deductive system for **FO(ID)** based on least fixpoint logic by introducing the logic **FO(ID,SLFP)** and extending Gentzen's sequent calculus for first

order logic to the proof system **LFO(ID,SLFP)** for FO(ID,SLFP). The main technical results are the soundness theorem of **LFO(ID,SLFP)** for a slightly restricted fragment of FO(ID) and the completeness theorem of **LFO(ID,SLFP)** for a more restricted fragment of FO(ID).

Related work is provided by Brotherston in [3]. He introduced the language FOL_{ID} of first order logic with the schema for inductive definitions, which is based upon Martin-Löf's "ordinary production" [17] and developed a proof system which is sound and complete with respect to a standard model in FOL_{ID}. A similar work to Brotherston's is studied by Hagiya and Sakurai in [13]. They proposed to interpret a (stratified) logic program as iterated inductive definitions of Martin-Löf and developed a proof theory which is sound with respect to the perfect model, and hence, the well-founded semantics of logic programming. Actually, both the FOL_{ID} and the stratified logic programs as iterative definitions can be generalized in FO(ID). A formal proof system based on tableau methods for analyzing computation for Answer Set Programming (ASP) was given as well by Gebser and Schaub [11]. As shown in [15], ASP is closely related to FO(ID). Their approach furnishes declarative and fine-grained instruments for characterizing operations as well as strategies of ASP-solvers and provides a uniform proof-theoretic framework for analyzing and comparing different algorithms.

The first topic for future work is the development of tools to check the correctness of the outputs generated by FO(ID) model generators such as MiniSat(ID) [16]. Given an FO(ID) theory T and a finite domain D as input, a model generator outputs a model for T with domain D or concludes that T is unsatisfiable in D . In the former case, an independent *model checker* can be used to check whether the output is indeed a model of T . However, when the model generator concludes that T is unsatisfiable in D , it is less obvious how to check the correctness of this answer. A similar problem arises in SAT, where certain solvers can output a trace of a failed computation in the form of a resolution proof [21]. An independent *proof checker* can then be used to check this formal proof. For the future, we aim to study how to transform a trace of the computation of a model generator such as MiniSat(ID) into a proof of (a propositional version of) our proof system. Model and proof checkers can be a great help to detect bugs in model generators. Also we intend to develop semi-automated and automated theorem proving systems for FO(ID) from the current deductive system for FO(ID) based on the least fixpoint logic, which may have more applications and implementations.

References

1. C. Baral, G. Brewka, and J. S. Schlipf, editors. *Logic Programming and Nonmonotonic Reasoning, 9th International Conference, LPNMR 2007, Tempe, AZ, USA, May 15-17, 2007, Proceedings*, volume 4483 of *Lecture Notes in Computer Science*. Springer, 2007.
2. R.J. Brachman and H.J. Levesque. Competence in knowledge representation. In *National Conference on Artificial Intelligence (AAAI'82)*, pages 189–192, 1982.

3. J. Brotherston. *Sequent Calculus Proof System for Inductive Definitions*. PhD thesis, University of Edinburgh, 2006.
4. S.R. Buss. *Handbook of Proof Theory*. Elsevier, 1998.
5. K.J. Compton. Stratified least fixpoint logic. *Theoretical Computer Science*, 131(1):95–120, 1994.
6. M. Denecker. The well-founded semantics is the principle of inductive definition. In J. Dix, L. Fariñas del Cerro, and U. Furbach, editors, *Logics in Artificial Intelligence (JELIA '98)*, volume 1489 of *Lecture Notes in Artificial Intelligence*, pages 1–16. Springer-Verlag, 1998.
7. M. Denecker, M. Bruynooghe, and V. Marek. Logic programming revisited: Logic programs as inductive definitions. *ACM Transactions on Computational Logic (TOCL)*, 2(4):623–654, October 2001.
8. M. Denecker and E. Ternovska. Inductive Situation Calculus. In *Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 545–553, 2004.
9. M. Denecker and E. Ternovska. A logic of non-monotone inductive definitions. *ACM Transactions On Computational Logic (TOCL)*, 9(2), March 2008.
10. M. Denecker and J. Vennekens. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In Baral et al. [1], pages 84–96.
11. M. Gebser and T. Schaub. Tableau calculi for answer set programming. In S. Etalle and M. Truszczynski, editors, *Logic Programming, 22nd International Conference, ICLP 2006*, volume 4079 of *Lecture Notes in Computer Science*, pages 11–25. Springer, 2006.
12. G. Gentzen. Untersuchungen über das logische schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.
13. M. Hagiya and T. Sakurai. Foundation of Logic Programming Based on Inductive Definition. *New Generation Computing*, 2:59–77, 1984.
14. P. Hou, J. Wittocx, and M. Denecker. A deductive system for PC(ID). In Baral et al. [1], pages 162–174.
15. M. Mariën, D. Gilis, and M. Denecker. On the relation between ID-Logic and Answer Set Programming. In J. J. Alferes and J. A. Leite, editors, *JELIA*, volume 3229 of *Lecture Notes in Computer Science*, pages 108–120. Springer, 2004.
16. M. Mariën, J. Wittocx, M. Denecker, and M. Bruynooghe. SAT(ID): Satisfiability of propositional logic extended with inductive definitions. In *Proceedings of the Eleventh International Conference on Theory and Applications of Satisfiability Testing, SAT 2008*, *Lecture Notes in Computer Science*, pages 211–224. Springer, 2008.
17. P. Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In J.e. Fenstad, editor, *Second Scandinavian Logic Symposium*, pages 179–216, 1971.
18. M.E. Szabo, editor. *The Collected Papers of Gerhard Gentzen*. North-Holland Publishing Co., Amsterdam, 1969.
19. A. Van Gelder, K.A. Ross, and J.S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.
20. J. Vennekens and M. Denecker. FO(ID) as an extension of DL with rules. In *The 6th Annual European Semantic Web Conference*, 2009. Accepted.
21. Lintao Zhang and Sharad Malik. Validating sat solvers using an independent resolution-based checker: Practical implementations and other applications. In *2003 Design, Automation and Test in Europe Conference and Exposition (DATE 2003), 3-7 March 2003, Munich, Germany*, pages 10880–10885. IEEE Computer Society, 2003.