Sulla decidibilità di programmi FDNC
On the decidability of FDNC programs

P. A. Bonatti

SOMMARIO/*ABSTRACT*

Questo articolo introduce una nuova dimostrazione della decidibilità del controllo di consistenza per i programmi FDNC sotto la semantica dei modelli stabili, basandosi su *splitting sequences* regolari. Con questa tecnica, riusciamo a rilassare leggermente la definizione di programmi FDNC e muoviamo un primo passo verso l'analisi delle relazioni tra programmi FDNC e la teoria dei programmi finitamente ricorsivi.

We provide a new decidability proof for the consistency of FDNC programs under the stable model semantics, based on regular splitting sequences. With this technique, we can slightly relax the definition of FDNC programs and make a first step towards a precise understanding of the relationships between FDNC programs and finitely recursive programs.

Keywords: Answer set programming, finitely recursive, finitary, and FDNC programs, module sequences.

1 Introduction

Some of the recent works of Alberto concern modal extensions of logic programming [5, 1]. A major motivation for those programs is reasoning about actions and change. In this setting, nonmonotonic constructs such as negation as failure are extremely useful to encode compactly the frame axiom and action consequences. However, for a long time such features could be supported only by forbidding function symbols, in order to ensure decidability.

Later results dropped this restriction. *Finitary programs* [4] preserve decidability even in the presence of infinite domains. This is achieved at the cost of restrictions on the cycles in dependency graphs containing an odd number of negative edges. Such limitations imply restrictions on the constraints (in the form of *denials* like $\leftarrow A_1, \ldots, A_n$, for example) that can be encoded in a finitary program.

FDNC programs [8] adopt a different strategy. They restrict the syntax to (a skolemized form of) 2-variable guarded logic and avoid the restrictions on cycles and constraints.

In this paper we reformulate the decidability of the consistency check for FDNC programs in terms of regular splitting sequences. In this way we slightly generalize a decidability result published in [8].

2 Preliminaries

We assume the reader to be familiar with Logic Programming and the stable model semantics [2]

Disjunctive logic programs are sets of (disjunctive) rules

$$A_1 \vee A_2 \vee ... \vee A_m \leftarrow L_1, ..., L_n \quad (m > 0, n \geq 0),$$

where each A_j $(j = 1, ..., m)$ is a logical atom and each L_i $(i = 1, ..., n)$ is a *literal*, that is, either a logical atom A or a negated atom $\mathtt{not}\ A$.

If r is a rule with the above structure, then let $head(r) = \{A_1,\ A_2, ...,\ A_m\}$ and $body(r) = \{L_1, ...,\ L_n\}$. Moreover, let $body^+(r)$ (respectively $body^-(r)$) be the set of all atoms A s.t. A (respectively $\mathtt{not}\ A$) belongs to $body(r)$.

The ground instantiation of a program P is denoted by $Ground(P)$, and the set of atoms occurring in $Ground(P)$ is denoted by $atom(P)$. Similarly, $atom(r)$ denotes the set of atoms occurring in a rule r.

A Herbrand model M of P is a *stable model* of P iff $M \in \mathsf{lm}(P^M)$, where $\mathsf{lm}(X)$ denotes the set of least models of a positive (possibly disjunctive) program X, and P^M is the *Gelfond-Lifschitz transformation* of P, obtained from $Ground(P)$ by (i) removing all rules r such that $body^-(r) \cap M \neq \emptyset$, and (ii) removing all negative literals from the body of the remaining rules.

Disjunctive programs may have one, none, or multiple stable models. We say that a program is *consistent* if it has at least one stable model; otherwise the program is *inconsistent*. A *skeptical* consequence of a program P is any formula satisfied by all the stable models of P. A *credulous* consequence of P is any formula satisfied by at least one stable model of P.

The *dependency graph of a program* P is a labelled directed graph, denoted by $DG(P)$, whose vertices are the ground atoms of P's language. Moreover, there exists an edge from A to B iff for some rule $r \in Ground(P)$, $A \in head(r)$ and either B occurs in $body(r)$, or $B \in head(r)$.

An atom A *depends* on B if there is a directed path from A to B in the dependency graph.

A disjunctive program P is *finitely recursive* [4, 3] iff each ground atom A depends on finitely many ground atoms in $DG(P)$.

A *FDNC program* is a set of disjunctive rules conforming to any of the following schemata:

(R1) $A_1(x) \vee ... \vee A_n(x) \leftarrow (\texttt{not})B_0(x), ..., (\texttt{not})B_l(x)$
(R2) $R_1(x,y) \vee ... \vee R_n(x,y) \leftarrow$ $(\texttt{not})P_0(x,y), \ldots, (\texttt{not})P_l(x,y)$
(R3) $R_1(x,f_1(x)) \vee ... \vee R_n(x,f_n(x)) \leftarrow$ $(\texttt{not})P_0(x,g_0(x)), \ldots, (\texttt{not})P_l(x,g_l(x))$
(R4) $A_1(y) \vee ... \vee A_n(y) \leftarrow$ $(\texttt{not})B_0(Z_0), ..., (\texttt{not})B_l(Z_l), R(x,y)$
(R5) $A_1(f(x)) \vee ... \vee A_n(f(x)) \leftarrow$ $(\texttt{not})B_0(W_0), ..., (\texttt{not})B_l(W_l), R(x,f(x))$
(R6) $R_1(x,f_1(x)) \vee ... \vee R_n(x,f_n(x)) \leftarrow$ $(\texttt{not})B_0(x), ..., (\texttt{not})B_l(x)$
(R7) $C_1(\vec{c}_1) \vee ... \vee C_n(\vec{c}_n) \leftarrow (\texttt{not})D_1(\vec{d}_1), ..., (\texttt{not})D_l(\vec{d}_l)$

where $n,l \geq 0$, $Z_i \in \{x,y\}$, $W_i \in \{x,f(x)\}$, and each $\vec{c}_i$, $\vec{d}_i$ is a tuple of one or two constants. Each rule r must be *safe*, i.e., each variable must occur in $body^+(r)$. Moreover at least one rule of type (R7) must be a fact.

Our results depend on a *splitting theorem* that allows to construct stable models in stages. In turn, this theorem is based on the notion of *splitting set* of a program P [2],[6], that is, any set U of atoms such that, for all rules $r \in Ground(P)$, if $head(r) \cap U \neq \emptyset$ then $atom(r) \subseteq U$. The set of rules $r \in Ground(P)$ such that $head(r) \cap U \neq \emptyset$ is called the *bottom* of P relative to the splitting set U and is denoted by $bot_U(P)$.

The partially evaluated complement of the bottom program determines the rest of each stable model. The *partial evaluation* of a ground logic program P with splitting set U w.r.t. a set of ground atoms X is the program $e_U(P,X)$ defined as follows:

$$\begin{aligned} e_U(P,X) = \{\, r' \mid\ & \textit{there exists } r \in P \textit{ s.t.} \\ & (body^+(r) \cap U) \subseteq X,\ (body^-(r) \cap U) \cap X = \emptyset, \\ & head(r') = head(r),\ body^+(r') = body^+(r) \setminus U, \\ & body^-(r') = body^-(r) \setminus U \,\}\,. \end{aligned}$$

We are finally ready to formulate the splitting theorem.

Theorem 1 (Splitting theorem [6]) *Let U be a splitting set for a logic program P. An interpretation M is a stable model of P iff $M = J \cup I$, where*

1. *I is a stable model of $bot_U(P)$, and*
2. *J is a stable model of $e_U(Ground(P) \setminus bot_U(P), I)$.*

The splitting theorem has been extended to *transfinite sequences* in [7]. A (transfinite) sequence is a family whose index set is an initial segment of ordinals, $\{\alpha : \alpha < \mu\}$. The ordinal μ is the *length* of the sequence.

A sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of sets is *monotone* if $U_\alpha \subseteq U_\beta$ whenever $\alpha < \beta$, and *continuous* if, for each limit ordinal $\alpha < \mu$, $U_\alpha = \bigcup_{\nu<\alpha} U_\nu$. A sequence with $\mu = \omega$ is *smooth* if each of its elements is finite.

Definition 2 [Lifschitz-Turner, [7]] A *splitting sequence* for a program P is a monotone, continuous sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ of splitting sets for P s.t. $\bigcup_{\alpha<\mu} U_\alpha = atom(P)$.

Lifschitz and Turner generalized the splitting theorem to splitting sequences.

Theorem 3 (Splitting sequence theorem [7]) *Let P be a disjunctive program. M is a stable model of P iff there exists a splitting sequence $\langle U_\alpha \rangle_{\alpha<\mu}$ such that*

1. *M_0 is a stable model of $bot_{U_0}(P)$,*
2. *for all successor ordinals $\alpha < \mu$, M_α is a stable model of $e_{U_{\alpha-1}}(bot_{U_\alpha}(P) \setminus bot_{U_{\alpha-1}}(P), \bigcup_{\beta<\alpha} M_\beta)$,*
3. *for all limit ordinals $\lambda < \mu$, $M_\lambda = \emptyset$,*
4. *$M = \bigcup_{\alpha<\mu} U_\alpha$.*

3 Revised decidability results

We first observe that strictly speaking, FDNC programs are not always finitely recursive, due to the presence of *local variables*, i.e. variables that occur in the body of a rule and not in its head. Such variables arise in instances of rule schema (R4); in particular x occurs only in the body. However it is not hard to verify that the following proposition holds:

Proposition 4 *If an atom $R(t,u)$ belongs to a stable model of an FDNC program, then either $u = f(t)$ for some function symbol f, or (t,u) is one of the vectors of constants $\vec{c}_i$ occurring in the head of some instance of (R7).*

It follows that each rule of the form (R4) can be replaced by a finite number of its instances:

- one for each substitution $[y/f(x)]$, where f is a function symbol occurring in the program;
- one for each substitution $[x/a_1, y/a_2]$ for each vector of constants $\vec{c}_i = (a_1, a_2)$ occurring in the head of some instance of schema (R7).

By Proposition 4, such transformation preserves the set of stable models of the given FDNC program. Moreover, the transformation removes all local variables so the transformed program is a finitely recursive FDNC program.

With a similar argument we can further normalize FDNC programs, restricting the set of atoms that may occur in a rule head. Each instance of schema (R2) can be replaced by a finite number of its instances by analogy with the previous case. By Proposition 4, such transformation preserves the set of stable models of the given FDNC program. Moreover, the transformation specializes the heads of the instances of (R2) so that the following lemma holds:

Lemma 5 *Every FDNC program is equivalent to a FDNC program with no rules of the form* (R2) *or* (R4).

Corollary 6 *Every FDNC program P is equivalent to a finitely recursive FDNC program P' such that the binary atoms occurring in $Ground(P')$ are of the form $R(t, f(t))$ (for some function symbol f) or $R(\vec{c}_i)$, where $\vec{c}_i$ occurs in the head of some instance of* (R7).

Note that the above program transformation can be effectively computed. Therefore, from now on, we shall focus without loss of generality on *normal FDNC programs*, that we define as programs whose rules conform to some of the schemata (R1), (R3), (R5), (R6), and (R7).

In the following, let P be a given normal FDNC program, and let us construct a suitable splitting sequence for P. First take any effective enumeration $t_1, t_2, \ldots, t_i, \ldots$ of the ground compound terms of P's language, such that each term t_i precedes all the terms larger than t_i (in terms of the number of function symbol occurrences). For all such ground terms t_i, we shall denote by $\hat{U}_i$ the set of all ground atoms $A(t_i)$ and $R(t_i, f(t_i))$, for all function symbols f. Now a *canonical splitting sequence* for P can be defined as follows:

- let U_0 be the set of all atoms of the form $A(c)$, $R(c, d)$, or $R(c, f(c))$, where c and d are constants;
- let $U_{i+1} = U_i \cup \hat{U}_{i+1}$.

Since P has no rules conforming to (R2) or (R4), it is easy to check that $\langle U_i \rangle_{i<\omega}$ is indeed a splitting sequence for $Ground(P)$.

Moreover, note that by definition, canonical sequences are smooth, as U_0 and the sets $\hat{U}_i$ are all finite.

Another important property of canonical sequences is that the program slices $P_{i+1} = bot_{U_{i+1}}(P) \setminus bot_{U_i}(P)$ they induce are all isomorphic to each other. By isomorphic, we mean that for all $0 < i < j < \omega$, P_j can be obtained from P_i by uniformly replacing t_i with t_j (in symbols, $P_j = P_i[t_i/t_j]$).

Now consider finite sequences of models $\langle M_i \rangle_{i<k}$ with the following properties:

- M_0 is a stable model of $bot_{U_0}(P)$;
- M_{i+1} is a stable model of $e_{U_i}(bot_{U_{i+1}}(P) \setminus bot_{U_i}(P), M_i)$.

We say such a sequence is *blocked* if $M_k = M_j[t_j/t_k]$ for some j such that $1 < j < k$, that is, M_k can be obtained from M_j by replacing term t_j with t_k.

Lemma 7 *Every blocked model sequence $\langle M_i \rangle_{i<k}$ (induced by a canonical splitting sequence $\langle U_i \rangle_{i<\omega}$ for a normal FDNC program P) can be extended to an infinite sequence $\langle M_i \rangle_{i<\omega}$ satisfying the following properties:*

1. *M_0 is a stable model of $bot_{U_0}(P)$;*
2. *M_{i+1} is a stable model of $e_{U_i}(bot_{U_{i+1}}(P) \setminus bot_{U_i}(P), M_i)$.*

Roughly speaking, the idea simply consists in repeating the subsequence $M_j, \ldots, M_{k-1}$ forever, replacing the terms $t_j, \ldots, t_{k-1}$ as appropriate.

Proof. Let $\langle M_i \rangle_{i<k}$ be a blocked sequence as described in the lemma's statement. Point 1 follows immediately from the hypothesis, so we focus on point 2. Since $\langle M_i \rangle_{i<k}$ is blocked, there exists $j < k$ such that $M_k = M_j[t_j/t_k]$. For all $i > k$, let $m_i = j + (i - k) \bmod (k - j)$ and $M_i = M_{m_i}[t_{m_i}/t_i]$. Moreover, for all $i \geq 0$ let $P_{i+1} = bot_{U_{i+1}}(P) \setminus bot_{U_i}(P)$. As we already pointed out before this lemma, $P_i = P_{m_i}[t_{m_i}/t_i]$. Now, since both the program slices and the models with indexes i and m_i are subject to the same symbol renaming, we have that

$$e_{U_i}(P_i, M_{i-1}) = e_{U_{m_i}}(P_{m_i}, M_{m_i - 1})[t_{m_i}/t_i] \,.$$

Since semantics does not depend on symbol names and by assumption M_{m_i} is a stable model of $e_{U_{m_i}}(P_{m_i}, M_{m_i - 1})$ (as m_i lies between j and k), we clearly have that M_i is a stable model of $e_{U_i}(P_i, M_{i-1})$; this proves point 2. ∎

Now proving decidability is relatively easy. We start by characterizing satisfiability in terms of blocked sequences.

Theorem 8 *M is a stable model of a normal FDNC program P iff M is the limit of the extension (in the sense of Lemma 7) of a blocked sequence $\langle M_i \rangle_{i<k}$ (induced by a canonical splitting sequence $\langle U_i \rangle_{i<\omega}$ for P).*

Proof. (*If*) Suppose M is the limit of a sequence $\langle M_i \rangle_{i<\omega}$ such that $\langle M_i \rangle_{i<k}$ is a blocked sequence and such that:

1. M_0 is a stable model of $bot_{U_0}(P)$;
2. M_{i+1} is a stable model of $e_{U_i}(bot_{U_{i+1}}(P) \setminus bot_{U_i}(P), M_i)$.

Note that each program slice $P_{i+1} = bot_{U_{i+1}}(P) \setminus bot_{U_i}(P)$ contains only atoms from $U_{i+1} \setminus U_{i-1}$ (because P is a normal FDNC program). Therefore the partial evaluation of P_{i+1} does not depend on the atoms in U_{i-1}, that is, for all $i < \omega$,

$$e_{U_i}(P_{i+1}, \bigcup_{j \leq i} M_j) = e_{U_i}(P_{i+1}, M_i) \,.$$

Then properties 1 and 2 above entail the properties required by the splitting sequence theorem (for $\mu = \omega$). It follows that the limit $M = \bigcup_{i<\omega} M_i$ is a stable model of P.

(*Only if*) Suppose that M is a stable model of P. Let $M_0 = M \cap U_0$ and for all $i < \omega$, let $M_{i+1} = M \cap (U_{i+1} \setminus U_i)$. By the splitting theorem, M_0 is a stable model of $bot_{U_0}(P)$. Moreover, by applying the splitting theorem twice for all i, we have that each M_{i+1} is a stable model of $e_{U_i}(bot_{U_{i+1}}(P) \setminus bot_{U_i}(P), \bigcup_{j\leq i} M_j)$ that, as we pointed out in the *If* part of the proof, equals $e_{U_i}(bot_{U_{i+1}}(P) \setminus bot_{U_i}(P), M_i)$. Then we are only left to show that the sequence $\langle M_i\rangle_{i<\omega}$ contains a blocked prefix $\langle M_i\rangle_{i<k}$, that is, for some j and k such that $0 < j < k < \omega$, $M_k = M_j[t_j/t_k]$.

To see this, observe that by definition for all $i > 0$, M_i is a subset of $\hat{U}_i$, and $\hat{U}_i$ is isomorphic to $\hat{U}_1$, that is, $\hat{U}_i = \hat{U}_1[t_1/t_i]$ and $|\hat{U}_i| = |\hat{U}_1|$. It follows that for all $i > 0$ there exists $S_i \subseteq \hat{U}_1$ such that $M_i = S_i[t_1/t_i]$. Since $\hat{U}_1$ is finite, there must be two indexes j and k and a set $S \subseteq \hat{U}_1$ such that $0 < j < k \leq 2^{|\hat{U}_1|}$, $M_j = S[t_1/t_j]$, and $M_k = S[t_1/t_k]$. Consequently, $M_k = S[t_1/t_j][t_j/t_k] = M_j[t_j/t_k]$, which completes the proof. ∎

Corollary 9 *A normal FDNC program P has a stable model iff P has a blocked model sequence $\langle M_i\rangle_{i<k}$ (induced by a canonical splitting sequence $\langle U_i\rangle_{i<\omega}$ for P) with $k \leq 2^{|\hat{U}_1|}$.*

Corollary 10 *Deciding whether a FDNC program P is consistent is decidable.*

Proof. Consistency can be nondeterministically checked as follows: First normalize P. Next for $i = 1, \dots, 2^{|\hat{U}_1|}$, build the program $e_{U_i}(P_{i+1}, M_i)$ and pick up one of its stable models M_{i+1}; if no such model exists, then return false. Check whether M_{i+1} is isomorphic to some previous M_j; if so, return true. Otherwise repeat the loop, or return false if the end of the loop is reached. Clearly, this algorithm returns true in at least one run iff P has a blocked model sequence $\langle M_i\rangle_{i<k}$ with $k \leq 2^{|\hat{U}_1|}$. By the above corollary, it follows that the algorithm solves the consistency problem for P. ∎

Our results do not need all the restrictions placed on FDNC programs. Proposition 4 holds even when the program is not safe, provided that the rules conforming to (R2) have nonempty bodies. The other proofs do not depend on safeness. In this sense, our results are slightly more general than those in [8].

4 Conclusions and future work

We have given an alternative proof of a decidability result of [8] for FDNC programs by proving that a consistent normal FDNC program has always a stable model which is the limit of a regular sequence $\langle M_i\rangle_{i<\omega}$ of stable models of the finite programs $e_{U_i}(P_{i+1}, M_i)$. Such a regular sequence can be finitely represented by a blocked sequence $\langle M_i\rangle_{i<k}$.

The term *blocked* is deliberately inspired by the notion of blocking in tableaux for modal and description logics. The intuitions in all these areas are analogous, and the goals are the same, namely decidable reasoning in the presence of infinite domains through a finite representation of infinite regular models.

We are planning to complete this investigation by characterizing credulous and skeptical reasoning and their computational complexity in terms of blocked model sequences. In particular, in order to provide effective reasoning methods, we are going to exploit the fact that normal FDNC programs are finitely recursive; for such programs the sequence of bottom programs induced by a smooth splitting ω-sequences is consistent iff the entire program is consistent. The consistency of the bottoms can be proved by (a suitable adaptation of) Lemma 7.

It will be interesting to inspect applications of these ideas to modal extensions of logic programming, in the spirit of [5], possibly exploiting the translation in [1].

REFERENCES

[1] M. Baldoni, L. Giordano, and A. Martelli. Translating a modal language with embedded implication into horn clause logic. In *ELP*, volume 1050 of *LNCS*, pages 19–33. Springer, 1996.

[2] C. Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, Cambridge, 2003.

[3] S. Baselice, P.A. Bonatti, and G. Criscuolo. On finitely recursive programs. In *ICLP 2007*, volume 4670 of *LNCS*, pages 89–103. Springer, 2007.

[4] Piero A. Bonatti. Reasoning with infinite stable models. *Artif. Intell.*, 156(1):75–111, 2004.

[5] Laura Giordano, Alberto Martelli, and Camilla Schwind. Ramification and causality in a modal action logic. *J. Log. Comput.*, 10(5):625–662, 2000.

[6] V. Lifschitz and H. Turner. Splitting a Logic Program. In *Proc. of the 12th Int. Conf. on Logic Programming*, MIT Press Series Logic Program, pages 581–595. MIT Press, 1995.

[7] Vladimir Lifschitz and Hudson Turner. Splitting a logic program. In *International Conference on Logic Programming*, pages 23–37, 1994.

[8] M. Simkus and T. Eiter. FDNC: Decidable nonmonotonic disjunctive logic programs with function symbols. In *LPAR 2007*, volume 4790 of *Lecture Notes in Computer Science*, pages 514–530. Springer, 2007.