

On the equivalence between CP-logic and LPADs

Joost Vennekens Marc Denecker
Maurice Bruynooghe

Report CW 436, April 2008



Katholieke Universiteit Leuven
Department of Computer Science

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

On the equivalence between CP-logic and LPADs

Joost Vennekens Marc Denecker
Maurice Bruynooghe

Report CW436, April 2008

Department of Computer Science, K.U.Leuven

Abstract

We give a detailed proof of the fact that the probabilistic logics of Logic Programs with Annotated Disjunctions (LPADs) and CP-logic are equivalent.

This report contains a detailed proof of the fact that Logic Programs with Annotated Disjunctions (LPADs) (6) and CP-logic (5) are equivalent. Before moving on to this proof, we first present some preliminaries from lattice theory and logic programming, and summarize the definition of LPADs and CP-logic.

1 Preliminaries

This section recalls some well-known definitions and results from lattice theory and logic programming.

1.1 Some concepts from lattice theory

A binary relation \leq on a set L is a *partial order* if it is reflexive, transitive and anti-symmetric. A partially ordered set $\langle L, \leq \rangle$ is a *lattice* if every pair (x, y) of elements of L has a unique least upper bound and greatest lower bound. Such a lattice $\langle L, \leq \rangle$ is *complete* if every non-empty subset $S \subseteq L$ has a least upper bound and greatest lower bound. A complete lattice has a least element \perp and a greatest element \top . An operator $O : L \rightarrow L$ is *monotone* if for every $x \leq y$, $O(x) \leq O(y)$. An element $x \in L$ is a *prefixpoint* of O if $x \geq O(x)$, a *fixpoint* if $x = O(x)$ and a *postfixpoint* if $x \leq O(x)$. If O is a monotone operator on a complete lattice, then for every postfixpoint y , there exists a least element in the set of all prefixpoints x of O for which $x \geq y$. This least prefixpoint greater than y of O is also the least fixpoint greater than y of O . Moreover, it can be constructed by successively applying O to y , i.e., as the limit of the sequence $(y, O(y), O(O(y)), \dots)$. In particular, because \perp is a trivial postfixpoint, O has a least prefixpoint which is equal to its least fixpoint and which can be constructed by successive application of O to \perp .

1.2 Some concepts from logic programming

We assume familiarity with classical logic. A *Herbrand* interpretation for an alphabet Σ is an interpretation, which has as its domain the set $HU(\Sigma)$ of all ground terms that can be constructed from Σ and which interprets each constants by itself and each function symbol f/n by the function mapping a tuple (t_1, \dots, t_n) to $f(t_1, \dots, t_n)$. We can identify a Herbrand interpretation with a set of ground atoms. A *partial* Herbrand interpretation is a function ν from the set $HB(\Sigma)$ of all ground atoms, also called the *Herbrand base*, to the set of truth values $\{\mathbf{f}, \mathbf{u}, \mathbf{t}\}$. A (total) Herbrand interpretation corresponds to a partial Herbrand interpretation that does not include \mathbf{u} in its range. On the set of truth values, one defines the *precision order* :

$$\mathbf{u} \leq_p \mathbf{f} \text{ and } \mathbf{u} \leq_p \mathbf{t}$$

and the *truth order*:

$$\mathbf{f} \leq_t \mathbf{u} \leq_t \mathbf{t}.$$

These orders can be pointwise extended to partial Herbrand interpretations. Each totally ordered set S of partial Herbrand interpretations has a \leq_p -least upperbound de-

noted $\text{lub}_{\leq_p}(S)$. The three-valued truth function φ^ν for sentences φ and partial Herbrand interpretations ν is defined by induction:

- $p^\nu = \nu(p)$, for $p \in HB(\Sigma)$;
- $(\psi \wedge \varphi)^\nu = \text{Min}_{\leq_t}(\psi^\nu, \varphi^\nu)$;
- $(\forall x \varphi(x))^\nu = \text{Min}_{\leq_t}(\{\psi(t)^\nu \mid t \in HU(\Sigma)\})$.
- $(\neg\varphi)^\nu = (\varphi^\nu)^{-1}$ where $\mathbf{f}^{-1} = \mathbf{t}$, $\mathbf{t}^{-1} = \mathbf{f}$, $\mathbf{u}^{-1} = \mathbf{u}$.

A crucial monotonicity property of three-valued logic is that $\nu \leq_p \nu'$ implies $\varphi^\nu \leq_p \varphi^{\nu'}$.

The well-founded semantics of logic programs was originally defined in (4). We present the formalization of this semantics in (2). Formally, a logic program P is a set of rules of the form $p \leftarrow \varphi$, where p is a ground atom and φ is a first-order sentence. We define a *well-founded induction* of P as a sequence of partial Herbrand interpretations $(\nu^\alpha)_{0 \leq \alpha \leq \beta}$ satisfying the following conditions:

- $\nu^0 = \perp_{\leq_p}$, the mapping of all atoms to \mathbf{u} ;
- $\nu^\lambda = \text{lub}_{\leq_p}(\{\nu^\beta \mid \beta < \lambda\})$, for each limit ordinal λ ;
- $\nu^{\alpha+1}$ relates to ν^α in one of the following ways:
 - $\nu^{\alpha+1} = \nu^\alpha[p : \mathbf{t}]$ such that for some rule $p \leftarrow \varphi \in P$, $\varphi^{\nu^\alpha} = \mathbf{t}$;
 - $\nu^{\alpha+1} = \nu^\alpha[U : \mathbf{f}]$ where U is an *unfounded set*, i.e., a set of ground atoms such that for each $p \in U$, $\nu^\alpha(p) = \mathbf{u}$ and for each rule $p \leftarrow \varphi \in P$, $\varphi^{\nu^{\alpha+1}} = \mathbf{f}$.

A well-founded induction is a sequence of increasing precision. We call a well-founded induction $(\nu^\alpha)_{\alpha \leq \beta}$ *terminal* if it cannot be extended with a strictly more precise interpretation. Each well-founded induction whose limit is a total interpretation is terminal. We now define the *well-founded model* of P as the limit of any such terminal well-founded induction. As the following result shows, this definition coincides with the standard one.

Theorem 1. (2) *Each terminal induction sequence of P converges to the well-founded model of P , as it was defined in (4).*

In certain logic programming variants, such as abductive logic programs (3) and ID-logic (1), a distinction is made between predicates that are *defined* by the program and predicates that are left *open*. The set of defined predicates must contain at least those predicates that appear in the head of a rule of the program. This distinction is similar to that between endogenous and exogenous predicates, which is common in probabilistic modeling. It is straightforward to generalize the well-founded semantics to this case. Given an interpretation O of the open predicates, we define a *well-founded induction of P in O* by the same inductive definition as for ordinary well-founded inductions, only we now have as a base case that ν^0 should be the least precise partial

Herbrand interpretation *that extends* O . It is easy to see that each ν^i in such a well-founded induction in O in fact extends O and also that it there are no open predicates, this definition simply coincides with the original one. The *well-founded model of P in O* is then the limit of any terminal well-founded induction of P in O .

1.3 CP-logic

We assume a logical vocabulary, whose predicates have been divided into endogenous and exogenous predicates.

A *CP-event* is then a statement of the following form:

$$(p_1 : \alpha_1) \vee \cdots \vee (p_n : \alpha_n) \leftarrow \varphi, \quad (1)$$

where φ is a first-order sentence, the p_i are ground atoms with an endogenous predicate and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

A *CP-theory* is a finite multiset of CP-events. For now, we will restrict attention to CP-theories in which all sentences φ are positive formulas, i.e., they do not contain negation.

We now define some notation to refer to different components of a CP-theory. The head $head(r)$ of a rule r of form (1) is the set of all pairs (p_i, α_i) appearing in the description of the effects of the event; the body of r , $body(r)$, is its precondition φ . By $head_{At}(r)$ we denote the set of all ground atoms p_i for which there exists an α_i such that $(p_i, \alpha_i) \in head(r)$. Similarly, by $body_{At}(r)$ we will denote the set of all ground atoms p which “appear”¹ in $body(r)$.

We will call a CP-event $E \leftarrow \varphi$ a *rule* if we want to emphasize that we are referring to a syntactical construct. We also introduce the concept of a *non-ground* rule as a way of concisely representing sets of CP-events with identical structure. Concretely, such a non-ground rule is of the form:

$$\forall \mathbf{x} (A_1 : \alpha_1) \vee \cdots \vee (A_n : \alpha_n) \leftarrow \varphi,$$

where the atoms A_i and the formula φ now may contain free variables, taken from the universally quantified tuple of variables \mathbf{x} . Such a non-ground rule is seen as an abbreviation for the set of all rules $r[\mathbf{x}/\mathbf{t}]$ that result from replacing the variables \mathbf{x} by a tuple \mathbf{t} of ground terms in alphabet Σ .

The semantics of a CP-theory is defined in terms of a tree structure, in which the edges are labeled with probabilities. Each node in this tree corresponds to a state of the domain, with the root representing its initial state and the leaves its possible final

¹More formally, we use $body_{At}(r)$ to denote $At(body(r))$, where At is the mapping from sentences to sets of ground atoms, that is inductively defined by:

- For $p(\mathbf{t})$ a ground atom, $At(p(\mathbf{t})) = \{p(\mathbf{t})\}$;
- For $\varphi \circ \psi$, with \circ either \vee or \wedge , $At(\varphi \circ \psi)$ is defined as $At(\varphi) \cup At(\psi)$;
- For $\neg\varphi$, $At(\neg\varphi) = At(\varphi)$;
- For $\Theta x \varphi$, with Θ either \forall or \exists , $At(\Theta x \varphi) = \cup_{t \in H_U(\Sigma)} At(\varphi[x/t])$, where $H_U(\Sigma)$ denotes the Herbrand universe for the alphabet Σ .

states. Formally, we will assume a function \mathcal{I} that maps each node s to a Herbrand interpretation $\mathcal{I}(s)$, which represents the state of the domain to which this node corresponds.

Definition 1. Let Σ be a vocabulary. A *probabilistic Σ -process* \mathcal{T} is a pair $\langle T; \mathcal{I} \rangle$, such that:

- T is a tree structure, in which each edge is labeled with a probability, such that for every non-leaf node s , the probabilities of all edges leaving s sum up to precisely 1;
- \mathcal{I} is a mapping from nodes of T to Herbrand interpretations of Σ .

If we interpret the probability associated to an edge (s, s') as the probability of making a transition from s to s' and assume that all these transitions are probabilistically independent, then we can associate to each node s the probability $\mathcal{P}(s)$ of a random walk in the tree, starting from its root, passing through s . Indeed, for the root \perp of the tree, $\mathcal{P}(\perp) = 1$ and for each other node s , $\mathcal{P}(s) = \prod_i \alpha_i$ where the α_i are all the probabilities associated to edges on the path from the \perp to s . Essentially, the mapping \mathcal{P} contains all the information that is present in the labeling of the edges and vice versa. To ease notation, we will sometimes take the liberty of identifying a probabilistic Σ -process $\langle T; \mathcal{I} \rangle$ with the triple $\langle T; \mathcal{I}; \mathcal{P} \rangle$ and ignoring the labels on the edges of T .

Each probabilistic Σ -process now induces an obvious probability distribution over the states in which the domain described by Σ might end up.

Definition 2. Let Σ be a vocabulary and $\mathcal{T} = \langle T; \mathcal{I}; \mathcal{P} \rangle$ a probabilistic Σ -process. By $\pi_{\mathcal{T}}$ we denote the probability distribution that assigns to each Herbrand interpretation I of Σ the probability $\sum_{s \in L_{\mathcal{T}}(I)} \mathcal{P}(s)$, where $L_{\mathcal{T}}(I)$ is the set of all leaves s of T for which $\mathcal{I}(s) = I$.

Like any probability distribution over interpretations, such a $\pi_{\mathcal{T}}$ also defines a set of possible worlds, namely that consisting of all I for which $\pi_{\mathcal{T}}(I) > 0$. If all the probabilities $\mathcal{P}(s)$ are non-zero, then this is simply the set of all $\mathcal{I}(l)$ for which l is a leaf of \mathcal{T} .

We now to relate the transitions in such a probabilistic Σ -process to the events described by a CP-theory.

Definition 3. Let Σ be a vocabulary, C a CP-theory in this vocabulary and \mathcal{T} a probabilistic Σ -process. Let $r \in C$ be a CP-event of the form:

$$(p_1 : \alpha_1) \vee \cdots \vee (p_n : \alpha_n) \leftarrow \varphi.$$

We say that r *happens* in a node s of \mathcal{T} if s has $n + 1$ children s_1, \dots, s_{n+1} , such that:

- For all $1 \leq i \leq n$, $\mathcal{I}(s_i) = \mathcal{I}(s) \cup \{p_i\}$ and the probability of edge (s, s_i) is α_i ;
- For s_{n+1} , $\mathcal{I}(s_{n+1}) = \mathcal{I}(s)$ and the probability of the edge (s, s_{n+1}) is $1 - \sum_i \alpha_i$.

The semantics of CP-logic now requires that events happen only in certain conditions. A first requirement is of course that its precondition should hold. However, there is also a more involved condition, for which we need some more mathematical machinery. The basic idea is that an event should only happen after all events that might still affect the truth of its precondition have already happened, i.e., this precondition should not merely be *currently* true, but should in fact already be guaranteed to also remain true in all potential *future* states. This naturally leads to a three-valued logic, where we have truth values **t** (guaranteed to remain true), **f** (guaranteed to remain false), and **u** (still subject to change). Recall that a three-valued interpretation ν is a mapping from the ground atoms of our vocabulary to the set of truth values $\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$, which induces for each formula φ a truth value φ^ν .

Now, if our probabilistic process is in a state s , then the atoms of which we are already sure that they are true are precisely those in $\mathcal{I}(s)$. To figure out which atoms are still unknown, we need to look at which events might still happen, i.e., at those rules r , for which $body(r)^\nu \neq \mathbf{f}$. Whenever we find such a rule, we know that the atoms in $head(r)$ might all still be caused and, as such, they must be at least unknown. We will now look at a derivation sequence, in which we start by assuming that everything that is currently not **t** is **f** and then gradually build up the set of unknown atoms by applying this principle. Formally, we consider a sequence $(\nu_i)_{0 \leq i \leq n}$ of three-valued interpretations. Initially, ν_0 assigns **f** to all atoms not in $\mathcal{I}(s)$. For each $i > 0$, there must be a rule r with $body(r)^{\nu_i} \neq \mathbf{f}$, such that, for all $p \in head(r)$ with $\nu_i(p) = \mathbf{f}$, it is the case that $\nu_{i+1}(p) = \mathbf{u}$, while for all other atoms p , $\nu_{i+1}(p) = \nu_i(p)$. A sequence that satisfies these properties is called a *hypothetical derivation sequence* in s . Such a sequence is *terminal* if it cannot be extended. A crucial property is now that all such sequences reach the same limit.

Theorem 2. *Every terminal hypothetical derivation sequence reaches the same limit, i.e., if $(\nu_i)_{0 \leq i \leq n}$ and $(\nu'_i)_{0 \leq i \leq m}$ are such sequences, then $\nu_n = \nu'_m$.*

For a state s in a probabilistic process, we will denote this unique limit as ν_s and refer to it as the *potential* in s . Such a ν_s now provides us with an estimate of which atoms might still be caused, given that we are already in state s .

We now formally define the semantics of CP-logic.

Definition 4 (Execution model). Let C be a positive CP-theory and X an interpretation of the exogenous predicates. A probabilistic Σ -process $\mathcal{T} = \langle T; \mathcal{I} \rangle$ is an *execution model* of C in context X , written $\mathcal{T} \models_X C$, iff there exists a mapping \mathcal{E} from the non-leaf nodes of \mathcal{T} to C , such that:

- For the root \perp of \mathcal{T} , $\mathcal{I}(\perp) = X$;
- In each non-leaf node s , an event $\mathcal{E}(s) \in \mathcal{R}_{\mathcal{E}}(s)$ happens, such that its precondition is satisfied in s , i.e., $\mathcal{I}(s) \models body(\mathcal{E}(s))$;
- For each leaf l of \mathcal{T} , there are no events $r \in \mathcal{R}_{\mathcal{E}}(s)$ for which $\mathcal{I}(l) \models body(r)$.
- For every node s , $(body(\mathcal{E}(s)))^{\nu_s} \neq \mathbf{u}$, with ν_s the potential in s .

If there are no exogenous predicates, we simply write $\mathcal{T} \models C$.

An important result is that all such execution models (if any exist), define the same probability distribution.

Theorem 3 (Uniqueness). *Let C be a CP-theory. If \mathcal{T}_1 and \mathcal{T}_2 are both execution models of C , then $\pi_{\mathcal{T}_1} = \pi_{\mathcal{T}_2}$.*

1.4 LPADs

In this section, we recall language of *Logic Programs with Annotated Disjunctions*, or *LPADs* for short. This is a probabilistic extension of logic programming, which is based on disjunctive logic programs. Formally, an LPAD is a set of rules:

$$(h_1 : \alpha_1) \vee \cdots \vee (h_n : \alpha_n) \leftarrow \varphi, \quad (2)$$

where the h_i are ground atoms and φ is a sentence. As such, LPADs are syntactically identical to CP-logic. However, we will define their semantics quite differently.

Every rule of form (2) represents a probability distribution over the following set of logic programming rules:

$$\{(h_i \leftarrow \varphi) \mid 1 \leq i \leq n\}.$$

From these distributions, a probability distribution over logic programs is then derived. To formally define this distribution, we introduce the following concept of a *selection*. We use the notation $head^*(r)$ to denote the set of pairs $head(r) \cup \{(\emptyset, 1 - \sum_{(h:\alpha) \in head(r)} \alpha)\}$, where \emptyset represents the possibility that none of the h_i 's are caused by the rule r .

Definition 5 (C-selection). Let C be an LPAD. A *C-selection* is a function σ from C to $\bigcup_{r \in C} head^*(r)$, such that for all $r \in C$, $\sigma(r) \in head^*(r)$. By $\sigma^h(r)$ and $\sigma^\alpha(r)$ we denote, respectively, the first and second element of the pair $\sigma(r)$. The set of all *C*-selections is denoted as \mathcal{S}_C .

The probability $P(\sigma)$ of a selection σ is now defined as $\prod_{r \in C} \sigma^\alpha(r)$. For a set $S \subseteq \mathcal{S}_C$ of selections, we define the probability $P(S)$ as $\sum_{\sigma \in S} P(\sigma)$. By C^σ we denote the logic program that consists of all rules $\sigma^h(r) \leftarrow body(r)$ for which $r \in C$ and $\sigma^h(r) \neq \emptyset$. Such a C^σ is called an *instance* of C . We will interpret these instances by the well-founded model semantics. Recall that, in general, the well-founded model of a program P , $wfm(P)$, is a pair (I, J) of interpretations, where I contains all atoms that are certainly true and J contains all atoms that might possibly be true. If $I = J$, then the well-founded model is called exact. Intuitively, if $wfm(P)$ is exact, then the truth of all atoms can be decided, i.e., everything that is not false can be derived. In the semantics of LPADs, we wanted to ensure that all uncertainty is expressed by means of the annotated disjunctions. In other words, given a specific selection, there should no longer be any uncertainty. We therefore impose the following criterion.

Definition 6 (Soundness). An LPAD C is *sound* iff all instances of C have an exact well-founded model.

For such LPADs, the following semantics can now be defined.

Definition 7 (Instance based semantics μ_C). Let C be a sound LPAD. For an interpretation I , we denote by $W(I)$ the set of all C -selections σ for which $wfm(C^\sigma) = (I, I)$. The *instance based semantics* μ_C of C is the probability distribution on interpretations, that assigns to each I the probability $P(W(I))$ of this set of selections $W(I)$.

It is straightforward to extend this definition to allow for exogenous predicates as well. Indeed, in Section 1.2, we have already seen how to define the well-founded semantics for rule sets with open predicates, and this is basically all that is needed. Concretely, given an interpretation X for a set of exogenous predicates, we can define the instance based semantics μ_C^X given X as the distribution that assigns, to each interpretation I of the endogenous predicates, the probability of the set of all selections σ for which (I, I) is the well-founded models of C^σ given X . Of course, this semantics is only defined for LPADs that are sound in X , meaning that the well-founded model of each C^σ given X is two-valued.

2 Proof of equivalence

This section will show that LPADs and CP-logic are equivalent.

We first prove some properties of the semantics of CP-logic. Let us consider a CP-theory C and state s in an execution model of C . We will denote by $\mathcal{R}(s)$ the set of all CP-events $r \in C$ that have not yet happened in s , i.e., for which there is no ancestor s' of s with $\mathcal{E}(s') = r$. Consider the collection O_s of all partial interpretations ν such that for each atom p , $p^\nu = \mathbf{t}$ iff $p^{\mathcal{I}(s)} = \mathbf{t}$, and for each rule $r \in \mathcal{R}(s)$, if $body(r)^\nu \neq \mathbf{f}$, then for each atom $p \in head_{At}(r)$, $p^\nu \neq \mathbf{f}$. Stated differently, ν can be obtained from $\mathcal{I}(s)$ by turning false atoms of ν into unknown atoms in such a way that if the body of some rule $r \in \mathcal{R}(s)$ is unknown or true in ν , then each of its head atoms is unknown or true in ν as well.

Proposition 1. *Let $(\nu_i)_{0 \leq i \leq n}$ be a hypothetical derivation sequence in state s .*

- For each $0 \leq i \leq n$ and each $\nu \in O_s$ it holds that $\nu \leq_p \nu_i$.
- The limit $\nu_n = \nu_s$ is an element of O_s .

Proof. The first property can be proven by a straightforward induction. Clearly, it holds that $\nu \leq_p \nu_0 = \mathcal{I}(s)$. Assume $\nu \leq_p \nu_i$ for some $i < n$. The true atoms of ν and ν_{i+1} are those of $\mathcal{I}(s)$, so they are the same. Therefore, it suffices to show that every atom p that is false in ν is also false in ν_{i+1} , or, since ν and ν_{i+1} have the same true atoms, that every such p is not unknown in ν_{i+1} . Assume towards contradiction that p is false in ν and unknown in ν_{i+1} . By the induction hypothesis, p is still false in ν_i . Therefore, p belongs to the head of some rule $r \in \mathcal{R}(s)$ such that $body(r)^{\nu_i} \neq \mathbf{f}$. Since $\nu \leq_p \nu_i$, this would imply that $body(r)^\nu \neq \mathbf{f}$, which, given that $\nu \in O_s$, leads to the contradiction that $p^\nu \neq \mathbf{f}$. Hence, p is false in ν_{i+1} . It follows that $\nu \leq_p \nu_{i+1}$.

As for the second property, it is clear that ν_s can be obtained from $\mathcal{I}(s)$ by turning some false atoms into unknown atoms, and that there are no more rules $r \in \mathcal{R}(s)$ with a non-false body and false atoms in the head w.r.t. ν_s . Hence, $\nu_s \in O_s$. \square

Let C be an LPAD. Let us define a *partial C-selection* as a partial function σ from C mapping rules r of a subset $\text{dom}(\sigma) \subseteq C$ to pairs $(p : \alpha) \in \text{head}^*(r)$. The probability function of selections can be extended to partial selections by setting $P(\sigma) = \prod_{r \in \text{dom}(\sigma)} \sigma^\alpha(r)$. Define also $S(\sigma)$ as the set of C -selections that extend σ . The following equation is obvious:

$$P(\sigma) = \sum_{\sigma' \in S(\sigma)} P(\sigma')$$

We define an *instance* of σ as any instance $C^{\sigma'}$ in which σ' is a C -selection that extends σ .

Let \mathcal{T} be an execution model of C . Clearly, each node s in \mathcal{T} determines a unique partial C -selection, denoted $\sigma(s)$. Formally, if $(s_i)_{0 \leq i \leq n}$ is the path from the root to s , then the domain of $\sigma(s)$ is $\{\mathcal{E}(s_i) \mid 0 \leq i < n\}$ and each rule $r = \mathcal{E}(s_i)$ in its domain is mapped to the atom $p \in \text{head}^*(r)$ that was selected for s_{i+1} . Moreover, we have

$$\mathcal{P}(s) = P(\sigma(s)) = \sum_{\sigma' \in S(\sigma(s))} P(\sigma'). \quad (3)$$

With the path $(s_i)_{0 \leq i \leq n}$ from the root to some node s , we now also associate a sequence of partial interpretations $(K_j)_{j=0}^{2n+1}$ defined as follows:

- $K_0 = \perp$, the partial interpretation mapping all atoms to \mathbf{u} .
- $K_{2i+1} = \nu_{s_i}$, for all $0 \leq i \leq n$.
- $K_{2i+2} = \nu_{s_i}[p : \mathbf{t}]$, for all $0 \leq i < n$, where p is the head atom of $\mathcal{E}(s_i)$ selected to obtain s_{i+1} .

Proposition 2. *For each $\sigma \in S(\sigma)$, $(K_j)_{j=0}^{2n+1}$ is a well-founded induction sequence of C^σ .*

Proof. The proof is by induction on the length n of the path from the root of \mathcal{T} to s .

We start by proving that $(K_j)_{j=0}^{2n}$ is a well-founded induction of all instances C^σ with $\sigma \in S(\sigma(s))$. If $n = 0$, then s is the root of the tree and $\sigma(s)$ is the empty partial selection. The sequence (K_0) is obviously a well-founded induction sequence of any instance C^σ . For $n > 0$, the induction hypothesis states that $(K_j)_{j=0}^{2n-1}$ is a well-founded induction sequence of all instance C^σ , where σ belongs to $S(\sigma(s_{n-1}))$. Let r be $\mathcal{E}(s_{n-1})$, the rule selected in s_{n-1} , and let $K_{2n} = K_{2n-1}[p : \mathbf{t}]$ where p was selected in the head of r to obtain s . Hence, $\text{body}(r)$ is true in $K_{2n-1} = \nu_{s_{n-1}}$. Clearly, for each $\sigma \in S(\sigma(s))$, C^σ contains the rule $p \leftarrow \text{body}(r)$. Consequently, $(K_j)_{j=0}^{2n}$ is a well-founded induction of C^σ .

Next, we prove that $(K_j)_{j=0}^{2n+1}$ is a well-founded induction of all C^σ with $\sigma \in S(\sigma(s))$. Let us investigate the set U of all atoms q such that $K_{2n}(q) \neq K_{2n+1}(q)$. We will prove that all atoms of U are unknown in K_{2n} and false in K_{2n+1} and that U is an unfounded set of C^σ . It then will follow that $(K_j)_{j=0}^{2n+1}$ is a well-founded induction of C^σ .

Let us first verify that all atoms in U are unknown in K_{2n} and false in K_{2n+1} . If $n = 0$, then $K_0 = \nu_0 = K_1$, so $U = \{\}$ and the statement trivially holds. Let

$n > 0$. Recall that K_{2n} is $\nu_{s_{n-1}}[p : \mathbf{t}]$, where p is the atom selected in the head of $\mathcal{E}(s_{n-1})$ to obtain s , and $K_{2n+1} = \nu_s$. It is easy to see that the true atoms of K_{2n} and K_{2n+1} are identical to those true in $\mathcal{I}(s)$. Hence, K_{2n} and K_{2n+1} only differ on false or unknown atoms. To show that U contains only atoms that are unknown in K_{2n} and false in K_{2n+1} , it therefore suffices to show that all atoms false in K_{2n} are also false in K_{2n+1} . To prove this, it suffices to show that $K_{2n} \in O_s$. Indeed, if $K_{2n} \in O_s$, Proposition 1 entails that $\nu_s \geq_p K_{2n}$ and hence, all atoms false in K_{2n} are false in $\nu_s = K_{2n+1}$.

We observe that, since $\nu_{s_{n-1}}$ belongs to $O_{s_{n-1}}$ (Proposition 1), all head atoms of rules $r \in \mathcal{R}(s_{n-1})$ with a non-false body in $\nu_{s_{n-1}}$, are true or unknown in $\nu_{s_{n-1}}$. In particular, $\mathcal{E}(s_{n-1}) \in \mathcal{R}(s_{n-1})$ and has a true body in $\nu_{s_{n-1}}$, hence p is true or unknown in $\nu_{s_{n-1}}$. It follows that:

$$\nu_{s_{n-1}} \leq_p \nu_{s_{n-1}}[p : \mathbf{t}] = K_{2n}.$$

It follows that any rule $r \in \mathcal{R}(s) \subseteq \mathcal{R}(s_{n-1})$ with a non-false body in K_{2n} , has a non-false body in $\nu_{s_{n-1}}$; hence, all atoms in the head of such an r are true or unknown in $\nu_{s_{n-1}}$ and, *a fortiori*, in $K_{2n} = \nu_{s_{n-1}}[p : \mathbf{t}]$. Thus, we obtain that $K_{2n} \in O_s$, as desired.

So far, we have proven that $K_{2n+1} = K_{2n}[U : \mathbf{f}]$ and that all elements in U are unknown in K_{2n} . It follows that $K_{2n} \leq_p K_{2n+1}$ and, more generally, that $K_j \leq_p K_{2n+1}$, for all $j \leq 2n$. All that remains to be shown is that U is an unfounded set of each instance of $\sigma(s)$. Let C' be such an instance and for any atom $q \in U$, let $q \leftarrow \varphi$ be a rule of C' . We need to show that φ is false in K_{2n+1} . The rule is obtained as an instance of some rule $r \in C$ with q in its head. The rule r is not one of the rules $\mathcal{E}(s_i)$ with $i < n$, since otherwise q would be true in $\mathcal{I}(s_j)$ for all $j > i$ and, in particular, also in $\nu_{s_n} = K_{2n+1}$, which would contradict the fact that we have already shown q to be false in K_{2n+1} . It follows that $r \in \mathcal{R}(s)$. Since $K_{2n+1} = \nu_s \in O_s$ and q is false in ν_s , $body(r) = \varphi$ is false in K_{2n+1} . □

Proposition 3. *For each leaf l of an execution model \mathcal{T} of C , $\mathcal{I}(l)$ is the well-founded model of each instance C^σ with $\sigma \in S(\sigma(l))$.*

Proof. Let l be a leaf and $\sigma \in S(\sigma(l))$. By Proposition 2, $(K_j)_{j=0}^{2n+1}$ is a well-founded induction of C^σ . Because l is a leaf, we have that for every rule $r \in \mathcal{E}(l)$, $body(r)$ is false in $\mathcal{I}(l)$. Therefore, $\mathcal{I}(l) \in O_l$ and Proposition 1 states that $\nu_l \geq_p \mathcal{I}(l)$. However, because $\mathcal{I}(l)$ is two-valued, this implies that $\nu_l = \mathcal{I}(l)$. Therefore, $K_{2n+1} = \nu_l$ is a total interpretation. Because a well-founded induction with a total limit is terminal, $\mathcal{I}(l)$ is the well-founded model of C^σ . □

This now allows us to prove the desired equivalence.

Theorem 4. *Let \mathcal{T} be an execution model of a CP-theory C . For each interpretation J ,*

$$\mu_C(J) = \pi_{\mathcal{T}}(J).$$

Proof. Given an execution model \mathcal{T} of a CP-theory C , we associate to each node s of \mathcal{T} the set $S(\sigma(s))$ of all those C -selections σ that extend $\sigma(s)$. It is easy to see that, with $L_{\mathcal{T}}$ the set of all leaves of \mathcal{T} , the class $\{S(\sigma(l)) \mid l \in L_{\mathcal{T}}\}$ is a partition of the set S_C of all selections. Let $L_{\mathcal{T}}(J)$ be the set of all leaves l of \mathcal{T} for which $\mathcal{I}(l) = J$. By Proposition 3, the well-founded model of a selection $\sigma \in S(\sigma(l))$ for a leaf l is $\mathcal{I}(l)$. Therefore, for a given two-valued interpretation J , the class $\{S(\sigma(l)) \mid l \in L_{\mathcal{T}}(J)\}$ is a partition of the collection $S(J)$ of selections σ such that $WFM(C^\sigma) = J$. This now allows us to derive the following equation:

$$\begin{aligned} \mu_C(J) = \sum_{\sigma \in S(J)} P(\sigma) &= \sum_{l \in L_{\mathcal{T}}(J)} \sum_{\sigma \in S(\sigma(l))} P(\sigma) \\ &= \sum_{l \in L_{\mathcal{T}}(J)} \mathcal{P}(l) && \text{(see equation (3))} \\ &= \pi_{\mathcal{T}}(J). \end{aligned}$$

□

References

- [1] Marc Denecker and Eugenia Ternovska. A logic of non-monotone inductive definitions. *Transactions On Computational Logic (TOCL)*, 2007.
- [2] Marc Denecker and Joost Vennekens. Well-founded semantics and the algebraic theory of non-monotone inductive definitions. In *Logic Programming and Non-monotonic Reasoning, 9th International Conference, LPNMR 2007, Proceedings*, volume 4483 of *Lecture Notes in Artificial Intelligence*, pages 84–96. Springer, 2007.
- [3] A. C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.
- [4] Allen Van Gelder, Kenneth Ross, and John Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, 1991.
- [5] J. Vennekens, M. Denecker, and M. Bruynooghe. Representing causal information about a probabilistic process. In *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Proceedings*, volume 4160 of *Lecture Notes in Computer Science*, page 452.
- [6] Joost Vennekens, Sofie Verbaeten, and Maurice Bruynooghe. Logic programs with annotated disjunctions. In *Logic Programming, 20th International Conference, ICLP 2004, Proceedings*, volume 3132 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 2004. URL = http://www.cs.kuleuven.ac.be/cgi-bin/dtai/publ_info.pl?id=41199.