Tractable Reasoning using Logic Programs with Intensional Concepts

Jesse Heyninck[1] [0000−0002−3825−4052], Ricardo Gonçalves[2] [0000−0003−4289−7197], Matthias Knorr[2] [0000−0003−1826−1498], and João Leite[2] [0000−0001−6786−7360]

[1] Technische Universität Dortmund jesse.heyninck@tu-dortmund.de
[2] NOVA LINCS, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa {rjrg,mkn,jleite}@fct.unl.pt

Abstract. Recent developments triggered by initiatives such as the Semantic Web, Linked Open Data, the Web of Things, and geographic information systems resulted in the wide and increasing availability of machine-processable data and knowledge in the form of data streams and knowledge bases. Applications building on such knowledge require reasoning with modal and intensional concepts, such as time, space, and obligations, that are defeasible. E.g., in the presence of data streams, conclusions may have to be revised due to newly arriving information. The current literature features a variety of domain-specific formalisms that allow for defeasible reasoning using specific intensional concepts. However, many of these formalisms are computationally intractable and limited to one of the mentioned application domains. In this paper, we define a general method for obtaining defeasible inferences over intensional concepts, and we study conditions under which these inferences are computable in polynomial time.

1 INTRODUCTION

In this paper, we develop a solution that allows us to tractably reason with intensional concepts, such as time, space and obligations, providing defeasible/non-monotonic inferences in the presence of large quantities of data.

Initiatives such as the Semantic Web, Linked Open Data, and the Web of Things, as well as modern Geographic Information Systems, resulted in the wide and increasing availability of machine-processable data and knowledge in the form of data streams and knowledge bases. To truly take advantage of this kind of knowledge, it is paramount to be able to reason in the presence of *intensional* or *modal* concepts, which has resulted in an increased interest in formalisms, often based on rules with defeasible inferences, that allow for reasoning with time [5,26,12,10,14,41], space [13,42,28,39], and possibility or obligations [36,25,27,11]. Examples of such concepts may be found in applications with data referring for example to time (e.g., operators such as “next”, “at time”, “during interval T”) or space (e.g., “at place P”, “within a given radius”, “connected to”), but also legal reasoning (e.g., “is obliged to”, “is permitted”).

Example 1. In a COVID-19-inspired setting, we consider an app for contact-tracing. It tracks where people move and stores their networks of persons, i.e., their colleagues and family whom they meet regularly. Once a person tests positive, the app informs anyone at risk (e.g. someone who was in the proximity of an infected person for a longer amount of time or because someone in their network is at risk) that they have to stay in quarantine for 10 days. If a negative test result can be given, this quarantine is not obligatory anymore. It is important that the app can explain to persons being orderd in quarantine the reason for their being at risk (e.g. since they were in contact with an infected person for a longer amount of time) while preserving anonymity to abide with laws of data protection (e.g. someone being ordered into quarantine should *not* be able to see who is the reason for this).

In this context, efficient reasoning with non-monotonic rules over intensional concepts is indeed mandatory, since a) rules allow us to encode monitoring and intervention guidelines and policies in a user-friendly and declarative manner; b) conclusions may have to be revised in the presence of newly arriving information; c) different intensional concepts need to be incorporated in the reasoning process; d) timely decisions are required, even in the presence of large amounts of data, as in streams; e) intensional concepts can preserve anonymity, e.g. in user-friendly explanations without having to change the rules. However, relevant existing work usually deals with only one kind of intensional concepts (as detailed before), and, in general, the computational complexity of the proposed formalisms is too high, usually due to both the adopted underlying formalism and the unrestricted reasoning with expressive intensional concepts.

In this paper, we introduce a formalism that allows us to seamlessly represent and reason with defeasible knowledge over different intensional concepts. We build on so-called intensional logic programs [34], extended with non-monotonic default negation, and equip them with a novel three-valued semantics with favorable properties. In particular, we define a well-founded model in the line of the well-founded semantics for logic programs [22]. Provided the adopted intensional operators satisfy certain properties, which turn out to be aligned with practical applications such as the one outlined in Ex. 1, the well-founded model is unique, minimal among the three-valued models, in the sense of only providing derivable consequences, and, crucially, its computation is tractable. Our approach allows us to add to relevant related work in the sense of providing a well-founded semantics to formalisms that did not have one so far, which we illustrate on a relevant fragment of LARS programs [10].

We introduce intensional logic programs in Sec. 2, define our three-valued semantics in Sec. 3, show how to compute the well-founded model in Sec. 4, discuss the complexity and related work in Secs. 5 and 6, respectively, before we conclude.

2 INTENSIONAL LOGIC PROGRAMS

In this section, building on previous work by Orgun and Wadge [34], we introduce intensional logic programs, a very expressive framework that allows us to reason with intensional concepts, such as time, space, and obligations, in the presence of large quantities of data, including streams of data. Intensional logic programs are based on rules, as used in normal logic programs, enriched with atoms that introduce the desired intensional concepts. The usage of default negation in the rules is a distinctive feature compared to the original work [34] and particularly well-suited to model non-monotonic and defeasible reasoning [23] and allows us to capture many other forms of non-monotonic reasoning, see, e.g., [16,19]. To assign meaning to intensional programs, we rely on the framework of neighborhood semantics [35], a generalization of the Kripke semantics, that easily allows us to capture a wide variety of intensional operators.

We start by defining the basic elements of our language. We consider a function-free first-order signature $\Sigma = \langle P, C \rangle$, a set X of variables, and a set of *operation symbols* $\mathcal{O}$, such that the sets P (of predicates), C (of constants), X and $\mathcal{O}$ are mutually disjoint. The set of atoms over Σ and X is defined in the usual way. We say that an atom is ground if it does not contain variables, and we denote by $\mathcal{A}_\Sigma$ the set of all ground atoms over Σ. In what follows, and without loss of generality, we leave the signature Σ implicit and consider only the set of ground atoms over Σ, denoted by $\mathcal{A}$.

The set $\mathcal{O}$ contains the symbols representing the various intensional operators ∇. Based on these, we introduce the set of intensional atoms $\mathcal{I}_\mathcal{O}^\mathcal{A}$.

Definition 1. *Given a set of atoms $\mathcal{A}$ and a set of operation symbols $\mathcal{O}$, the set $\mathcal{I}_\mathcal{O}^\mathcal{A}$ of* intensional atoms over $\mathcal{A}$ and $\mathcal{O}$ *is defined as $\mathcal{I}_\mathcal{O}^\mathcal{A} = \{\nabla p \mid p \in \mathcal{A}$ and $\nabla \in \mathcal{O}\}$[3], and the set of* program atoms $\mathcal{L}_\mathcal{O}^\mathcal{A}$ *is defined as $\mathcal{L}_\mathcal{O}^\mathcal{A} = \mathcal{A} \cup \mathcal{I}_\mathcal{O}^\mathcal{A}$.*

We can define intensional logic programs as sets of rules with default negation, denoted by $\sim$, over program atoms.

Definition 2. *Given a set of atoms $\mathcal{A}$ and a set of operation symbols $\mathcal{O}$, an* intensional logic program $\mathcal{P}$ over $\mathcal{A}$ and $\mathcal{O}$ *is a finite set of rules r of the form:*

$$A \leftarrow A_1, \ldots, A_n, \sim B_1, \ldots, \sim B_m \tag{1}$$

where $A, A_1, \ldots, A_n, B_1, \ldots, B_m \in \mathcal{L}_\mathcal{O}^\mathcal{A}$. We distinguish between the head *of r, A, and its* body*, $A_1, \ldots, A_n, \sim B_1, \ldots, \sim B_m$.*

We also call $\mathcal{P}$ simply a *program* when this does not cause confusion and *positive* if it does not contain default negation. Intensional logic programs are highly expressive as intensional operators can appear arbitrarily anywhere in the rules, in particular in rule heads and in scope of default negation.

Example 2. Let a set of agents $A = \{a, b, r\}$ (for Anita, Bonnie and Ruth) be given, a set of locations $L = \{\alpha, \beta, \gamma, \ldots\}$ and a set of time points $T = \{1, 2, \ldots, \}$. We also assume that every agent has a network $N_i \subseteq A$ which represents the people the agent has regular close contact with (e.g. family, colleagues or partner). In our example, $N_a = \{b\}$, $N_b = \{a\}$ and $N_r = \emptyset$. We furthermore assume a function $\nu : L \to \wp(L)$ which assigns to each place ℓ the places in its vicinity $\nu(\ell)$. In our example, for simplicity's sake, we just assume that $\nu(\alpha) = \{\beta\}$. We define the following operators for our use-case as $\mathcal{O}_1 = \{[i]_\ell, [i], [i]^t, [t, t'], \triangleright_t^i, \hat{\ell}^i, \langle A \rangle_\ell, \langle N_i \rangle \mid i \in A, \ell \in L, t \in T\}$ with the following informal interpretations: $[i]_\ell \phi$ says that ϕ is true for agent i at location ℓ; $[i]\phi$ says that ϕ is true for agent i; $[i]^t \phi$ says that ϕ is true for agent i at time t; $[t, t']\phi$ means that ϕ is the case in the interval between t and t'; $\triangleright_t^i \phi$ means ϕ is the case at or after

[3] For simplicity, we restrict ourselves to non-nested (or equivalently in view of Def. 2, composed) intensional atoms. This does not result in any loss of generality, since nested operators can straightforwardly be modelled as non-nested operators, see Remark 1.

time t for agent i; $\hat{\ell}^i\phi$ says that ϕ is true for an agent i in the vicinity of ℓ; $\langle A\rangle^t_\ell\phi$ says that ϕ is true for some agent $i \in A$ at location ℓ; and $\langle N_i\rangle\phi$ says that ϕ is true for some agent in i's network.

We use the atoms `risk`, `reside`, `inf`, `neg.test`, `quar`, and `spread`, which represent that someone is at risk of infection, is residing, is infected, has a negative test result, is imposed quarantine, and is a potential spreader, respectively. We can now, for example, succinctly write the following program (for any $i \in A$, $\ell \in L$ and $t \in T$):

$$\begin{aligned}
[i]_\ell\mathtt{spread} &\leftarrow [i]\mathtt{inf}, [i]_\ell\mathtt{reside}\\
[i]^t\mathtt{risk} &\leftarrow [t, t+x]\hat{\ell}^i\mathtt{reside}, [t, t+x]\langle A\rangle_\ell\mathtt{spread}\\
[i]\mathtt{risk} &\leftarrow \langle N_i\rangle\mathtt{risk}\\
[t, t+10][i]\mathtt{quar} &\leftarrow [i]^t\mathtt{risk}, \sim \triangleright^i_t\mathtt{neg.test}
\end{aligned}$$

These rules express that someone who is infected and resides at ℓ is a potential spreader at place ℓ; if agent i is in the vicinity of a potential spreader for at least x time units, i is at risk: if someone in agent i's network is at risk, so is i; if i is at risk at time t and does not have a negative test result after time t, i is imposed quarantine for the time between t and $t + 10$.

In order to give semantics to intensional operators, we follow the ideas employed by Orgun and Wadge [34] and consider the neighborhood semantics, a strict generalization of Kripke-style semantics that allows capturing intensional operators [35] such as temporal, spatial, or deontic operators, even those that do not satisfy the normality property imposed by Kripke frames [18]. We start by recalling neighborhood frames.

Definition 3. *Given a set of operation symbols $\mathcal{O}$, a* neighborhood frame *(over $\mathcal{O}$) is a pair $\mathfrak{F} = \langle W, N\rangle$ where W is a non-empty set (of worlds) and $N = \{\theta_\nabla \mid \nabla \in \mathcal{O}\}$ is a set of* neighborhood functions $\theta_\nabla : W \to \wp(\wp(W))$.[4]

Thus, in comparison to Kripke frames, instead of a relation over W, neighborhood frames have functions for each operator that map worlds to a set of sets of worlds. These sets intuitively represent the atoms necessary (according to the corresponding intensional operator) at that world.

Example 3. The operators from Ex. 2 are given semantics using a neighborhood frame. We define worlds $w \in \mathcal{W}$ as triples (i, ℓ, t) where $i \in A$, $\ell \in L$ and $t \in T$. These represent the space-time locations for an agent i.

The neighborhoods of $\mathcal{O}_1$ are defined, for $t, t', t^\star \in T$, $\ell, \ell' \in L$ and $i, i' \in A$:
- $\theta_{[i]_\ell}((i', \ell', t)) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i, \ell, t) \in \mathcal{W}'\}$.
- $\theta_{[i]^t}((i', \ell, t')) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i, \ell, t) \in \mathcal{W}'\}$.
- $\theta_{[i]}((i', \ell, t)) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i, \ell, t) \in \mathcal{W}'\}$.
- $\theta_{[t,t']}((i, \ell, t^\dagger)) = \{\mathcal{W}' \subseteq \mathcal{W} \mid \{(i, \ell, t^\star) \mid t^\star \in [t, t']\} \subseteq \mathcal{W}'\}$.
- $\theta_{\triangleleft^i_t}((i, \ell, t')) = \{\mathcal{W}' \subseteq \mathcal{W} \mid \{(i, \ell, t^\star) \mid \ell \in L\} \subseteq \mathcal{W}'$ for some $t^\star \leq t\}$.

[4] Note that we often leave $\mathcal{O}$ implicit as N allows to uniquely determine all elements from $\mathcal{O}$. Also, to ease the presentation, we only consider unary intensional operators. Others can then often be represented using rules (see also [34]).

– $\theta_{\hat{\ell}^i}((i',\ell',t)) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i,\ell^\star,t) \in \mathcal{W}'$ for some $\ell^\star \in \nu(\ell)\}$.
– $\theta_{\langle A\rangle_\ell}((i',\ell',t')) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i^\star,\ell,t) \in \mathcal{W}'$ for some $i^\star \in A\}$.
– $\theta_{\langle N_i\rangle_\ell^t}((i',\ell',t')) = \{\mathcal{W}' \subseteq \mathcal{W} \mid (i^\star,\ell,t) \in \mathcal{W}'$ for some $i^\star \in N_i\}$.

Intuitively, e.g., $\theta_{[i]_\ell}((i',\ell',t))$ consists of all the sets of worlds that include the world (i,ℓ,t) that shares a time component with the world (i',ℓ',t), but has ℓ and i as spatial and agent components; $\theta_{[t,t']^i}$ consists of all the sets of worlds that include all worlds $(i,\ell,t^\star)$ with some time component $t^\star$ between or equal to t and t' (for every place $\ell \in L$); and a set of worlds is contained in $\theta_{\langle A\rangle_\ell^t}$ if it contains at least one world with time component t and space component ℓ and some agent component i.

As the example above shows, neighborhood functions θ can be both invariant under the input w or variate depending on w (e.g., $\theta_{\langle A\rangle^{t,\ell}}$ and $\theta_{[t,t']^i}$ are invariant, while $\theta_{[i]_\ell}$ and $\theta_{[i]^t}$ variate depending on w). This is why the above definitions of neighborhood functions that depend on w need to explicit the components of the world w, i.e., (i,ℓ,t).

3 THREE-VALUED SEMANTICS

In this section, we define a three-valued semantics for intensional logic programs as an extension of the well-founded semantics for logic programs [22] that incorporates reasoning over intensional concepts. The benefit of this approach over the more commonly used two-valued models is that, although there are usually several such three-valued models, we can determine a unique minimal one – intuitively the one which contains all the minimally necessary consequences of a program – which can be efficiently computed. Recall that even for programs without intensional concepts, a unique two-valued minimal model does not usually exist [24].

We consider three truth values, "true", "false", and "undefined", where the latter corresponds to neither true nor false. Given a neighborhood frame, we start by defining interpretations that contain a valuation function which indicates in which worlds (of the frame) an atom from $\mathcal{A}$ is true ($W^\top$), and in which ones it is true or undefined (W^u), i.e., not false [5].

Definition 4. *Given a set of atoms $\mathcal{A}$ and a frame $\mathfrak{F} = \langle W, N\rangle$, an* interpretation I over $\mathcal{A}$ and $\mathfrak{F}$ *is a tuple $\langle W, N, V\rangle$ with a valuation function $V : \mathcal{A} \to \wp(W) \times \wp(W)$ s.t., for every $p \in \mathcal{A}$, $V(p) = (W^\top, W^u)$ with $W^\top \subseteq W^u$. If, for every $p \in \mathcal{A}$, $W^\top = W^u$, then we call I* total.

The subset inclusion on the worlds ensures that no $p \in \mathcal{A}$ can be true and false in some world simultaneously. This intuition of the meaning is made precise with the denotation of program atoms for which we use the three truth values. We denote the truth values true, undefined and false with $\top$, u, and $\bot$, respectively, and we assume that the language $\mathcal{L}_\mathcal{O}^\mathcal{A}$ contains a special atom u (associated to u).

Definition 5. *Given a set of atoms $\mathcal{A}$, a frame $\mathfrak{F}$, and an interpretation $I = \langle W, N, V\rangle$, we define the* denotation of $A \in \mathcal{L}_\mathcal{O}^\mathcal{A}$ in I*:*

[5] We follow the usual notation in modal logic and interpretations explicitly include the corresponding frame.

- $\|p\|_I^\dagger = W^\dagger$ *if* $A = p \in \mathcal{A}$*, with* $V(p) = (W^\top, W^u)$ *and* $\dagger \in \{\top, u\}$*;*
- $\|\mathsf{u}\|^u = W$ *and* $\|\mathsf{u}\|^\top = \emptyset$*, if* $A = \mathsf{u}$*;*
- $\|\nabla p\|_I^\dagger = \{w \in W \mid \|p\|_I^\dagger \in \theta_\nabla(w)\}$ *if* $A = \nabla p \in \mathcal{I}_\mathcal{O}^\mathcal{A}$ *and* $\dagger \in \{\top, u\}$*;*
- $\|A\|_I^\perp = W \setminus \|A\|_I^u$ *for* $A \in \mathcal{L}_\mathcal{O}^\mathcal{A}$.

For a formula $A \in \mathcal{L}_\mathcal{O}^\mathcal{A}$ and an interpretation I, $\|A\|_I^\top$ is the set of worlds in which A is true, $\|A\|_I^u$ is the set of worlds in which A is not false, i.e., undefined or true, and $\|A\|_I^\perp$ is the set of worlds in which A is false. For atoms $p \in \mathcal{A}$, the denotation is straightforwardly derived from the interpretation I, i.e., from the valuation function V, and for the special atom u it is defined as expected (undefined in all worlds). For an intensional atom ∇p, w is in the denotation $\|\nabla p\|_I^\dagger$ of ∇p if the denotation of p (according to I) is a neighborhood of ∇ for w, i.e. $\|p\|_I^\dagger \in \theta_\nabla(w)$.

We often leave the subscript I from $\|A\|_I^\dagger$ as well as the reference to $\mathcal{A}$ and $\mathfrak{F}$ for interpretations and programs implicit.

Example 4. Consider $I_1 = \langle W_1, \mathcal{O}_1, V\rangle$ with the set of worlds W_1 and the neighborhoods as in Ex. 3 where:

$V(\texttt{reside}) = (\{(a, \alpha, 1)\}, \{(a, \alpha, 1)\})$

Then the following are examples of denotations of intensional atoms:

$\|[a]^\alpha\texttt{reside}\|_{I_1}^\top = \{(i, \ell, 1) \mid i \in A, \ell \in L\}$

$\|\hat{\beta}^a\texttt{reside}\|_{I_1}^\top = \{(i, \ell, 1), (i, \ell, 2) \mid i \in A, \ell \in L\}$

We explain the first denotation $\|[a]^\alpha\texttt{reside}\|_{I_1}^\top$ as follows: since `reside` is true for agent a at α and time 1, $[a]^\alpha\texttt{reside}$ is true at every world with time stamp 1. More formally, this can be seen since the set of worlds in which `reside` is true is a neighborhood $\theta_{[a]^\alpha}$.

Based on the denotation, we can now define our model notion, which is inspired by partial stable models [38], which come with two favorable properties, minimality and support. The former captures the idea of minimal assumption, the latter provides traceable inferences from rules. We adapt this notion here by defining a reduct that, given an interpretation, transforms programs into positive ones, for which a satisfaction relation and a minimal model notion are defined.

Remark 1. Operators can be straightforwardly combined within our framework. Indeed, given two operators ∇_1 and ∇_2, the nesting of them, $\nabla_1\nabla_2$, can be seen as an operator $\nabla_1 \oplus \nabla_2$, where the neighborhood $\theta_{\nabla_2\oplus\nabla_1}(w)$ is defined as follows. First we define $\theta_\nabla^{-1} : \mathcal{W} \to \mathcal{W}$ as $\theta_\nabla^{-1}(\mathcal{W}') = \{w' \in \mathcal{W} \mid \mathcal{W}' \in \theta_\nabla(w')\}$. Intuitively, this is the set of worlds w' for which $\mathcal{W}'$ is a ∇-neighborhood of w', i.e. $w' \in \theta_\nabla^{-1}(\mathcal{W}')$ iff $\mathcal{W}' \in \theta_\nabla(w')$. We then define the neighborhood of the composition of ∇_1 and ∇_2 as:

$$\theta_{\nabla_2\oplus\nabla_1}(w) = \{\mathcal{W}' \subseteq \mathcal{W} \mid \theta_{\nabla_1}^{-1}(\mathcal{W}') \in \theta_{\nabla_2}(w)\}$$

It is not hard to see that for any $\phi \in \mathcal{A}$, $w \in \|\nabla_2 \oplus \nabla_1\phi\|^\dagger$ iff $\|\nabla_1\phi\|^\dagger \in \theta_{\nabla_2}(w)$ (for any $\dagger \in \top, u\}$). In other words, $\nabla_2 \oplus \nabla_1\phi$ is true at w iff the worlds at which $\nabla_1\phi$ is true is a neighborhood of ∇_2, as expected from a sound definition of nested operators.

Example 5. As an example of the neighborhood of a nesting of operators, consider $[t, t+x]\hat{\ell}^i$ as it occurs in the second rule of Example 2. Since $\theta^{-1}_{\hat{\ell}^i}(\mathcal{W}') = \{(i', \ell', t) \mid t \in T$ for which $(i, l^\star, t) \in \mathcal{W}'$ for some $l^\star \in \nu(l), \ell' \in L, i' \in A\}$, one can observe:

$$\theta_{[t,t+x]\hat{\ell}^i}(w) = \{\mathcal{W}' \subseteq \mathcal{W}_1 \mid \forall t^\star \in [t, t+x]\ \exists l^\star \in \nu(l) \text{ s.t. } (i, l^\star, t^\star) \in \mathcal{W}'\}$$

In other words, a formula $[t, t+x]\hat{\ell}^i\phi$ is true at (at world w) iff ϕ is true for agent i at some place $l^\star$ in the vicinity of l for every time point $t^\star$ within the inteval $[t, t+x]$.

We first adapt two orders for interpretations, the truth ordering, $\sqsubseteq$, and the knowledge ordering, $\sqsubseteq_k$. The former prefers higher truth values in the order $\bot < u < \top$, the latter more knowledge (i.e., less undefined knowledge). Formally, for interpretations I and I', and every $p \in \mathcal{A}$: $I \sqsubseteq I'$ iff $\|p\|^\dagger_I \subseteq \|p\|^\dagger_{I'}$ for every $\dagger \in \{\top, u\}$; $I \sqsubseteq_k I'$ iff $\|p\|^\top_I \subseteq \|p\|^\top_{I'}$ and $\|p\|^\bot_I \subseteq \|p\|^\bot_{I'}$. We write $I \prec I'$ if $I \preceq I'$ and $I' \not\preceq I$ for $\preceq \in \{\sqsubseteq, \sqsubseteq_k\}$.

We now generalize the notion of reduct to programs with intensional atoms.

Definition 6. *Let $\mathcal{A}$ be set of atoms, and $\mathfrak{F} = \langle W, N\rangle$ a frame. $\mathcal{P}/I_w$, the* reduct of a program $\mathcal{P}$ at $w \in W$ w.r.t. an interpretation I, *contains for each $r \in \mathcal{P}$ of the form (1):*

- $A \leftarrow A_1, \ldots, A_n$ *if* $w \notin \bigcup_{i \leq m} \|B_i\|^u$
- $A \leftarrow A_1, \ldots, A_n, \mathsf{u}$ *if* $w \in \bigcup_{i \leq m} \|B_i\|^u \setminus \bigcup_{i \leq m} \|B_i\|^\top$

Intuitively, for each rule r of $\mathcal{P}$, the reduct $\mathcal{P}/I_w$ contains either (a) a rule of the first form, if all negated program atoms in the body of r are false at w (or the body does not have negated atoms), or (b) a rule of the second form, if none of the negated program atoms in the body of r are true at w, but some of these are undefined at w, or (c) none, otherwise. This also explains why the reduct is defined at w: truth and undefinedness vary for different worlds. The special atom u is applied to ensure that rules for the second case cannot impose the truth of the head in the notion of satisfaction for positive programs.

As reducts are positive programs, we can define a notion of satisfaction as follows.

Definition 7. *Let $\mathcal{A}$ be a set of atoms, and $\mathfrak{F} = \langle W, N\rangle$ a frame. An interpretation I* satisfies *a positive program $\mathcal{P}$ at $w \in W$ iff for each $r \in \mathcal{P}$ of the form (1), we have that $w \in \bigcap_{i \leq n} \|A_i\|^\dagger$ implies $w \in \|A\|^\dagger$ (for any $\dagger \in \{\top, \mathsf{u}\}$)* [6].

Stable models can now be defined by imposing minimality w.r.t. the truth ordering on the corresponding reduct.

Definition 8. *Let $\mathcal{A}$ be set of atoms, and $\mathfrak{F} = \langle W, N\rangle$ a frame. An interpretation I is* a stable model *of a program $\mathcal{P}$ if:*

- *for every $w \in W$, I satisfies $\mathcal{P}/I_w$ at w, and*
- *there is no interpretation I' s.t. $I' \sqsubset I$ and, for each $w \in W$, I' satisfies $\mathcal{P}/I_w$ at w.*

[6] Since the intersection of an empty sequence of subsets of a set is the entire set, then, for n=0, i.e., when the body of the rule is empty, the satisfaction condition is just $w \in \|A\|^\dagger$ for any $\dagger \in \{\top, \mathsf{u}\}$.

Example 6. We consider the following program on the basis of $\mathcal{P}$ from Ex. 2, zooming in on the part restricted to considerations pertaining to the network of an agent (rules 3 and 4 of that example) and adding the information that Anita was at risk at place α on time 1. This results in the following program $\mathcal{P}'$:

$$[i]\mathtt{risk} \leftarrow \langle N_i \rangle \mathtt{risk} \quad [t,t+10]^i \mathtt{quar} \leftarrow [i]^t \mathtt{risk}, \sim \triangleright^i_t \mathtt{neg.test} \quad [a]^1_a \mathtt{risk} \leftarrow$$

Consider $\mathfrak{F} = \langle W_1, \mathcal{O}_1 \rangle$ as in Ex. 3 and the total interpretation I_1 defined by:

$$\|\mathtt{risk}\|^{\top}_{I_1} = \{(a,\alpha,1),(b,\alpha,1)\} \quad \|\mathtt{quar}\|^{\top}_{I_1} = \{(i,\alpha,t), \mid t \leq 10, i \in \{a,b\}\}$$
$$\|\mathtt{neg.test}\|^{\top}_{I_1} = \emptyset$$

We see that, for any $w \in W_1$, $\mathcal{P}'/(I_1)_w$ consists of the following rules:

$$[i]\mathtt{risk} \leftarrow \langle N_i \rangle \mathtt{risk} \qquad [t,t+10]^i \mathtt{quar} \leftarrow [i]^t \mathtt{risk} \qquad [a]^t_\ell \mathtt{risk} \leftarrow$$

It can be checked that I_1 satisfies minimality and is therefore a stable model of $\mathcal{P}$.

Consider now the following total interpretation I_2 defined by:

$$\|\mathtt{risk}\|^{\top}_{I_2} = \{(a,\alpha,1)\} \qquad \|\mathtt{quar}\|^{\top}_{I_2} = \emptyset \qquad \|\mathtt{neg.test}\|^{\top}_{I_2} = \emptyset$$

We see that for any $w \in W_1$, $\mathcal{P}'/(I_2)_w = \mathcal{P}'/(I_1)_w$. Notice that I_1 is not a stable model of $\mathcal{P}$, since for $(a,\alpha,1) \in \|\langle N_b \rangle \mathtt{risk}\|^{\top}_{I_1}$ yet $(a,\alpha,1) \notin \|[b]\mathtt{risk}\|^{\top}_{I_1}$, since $(b,\alpha,1) \notin \|\mathtt{risk}\|^{\top}_{I_1}$.

We can show that our model notion is faithful w.r.t. partial stable models of normal logic programs [38], i.e., if we consider a program without intensional atoms, then its semantics corresponds to that of partial stable models.

Proposition 1. *Let $\mathcal{A}$ be set of atoms, $\mathfrak{F}$ a frame, and $\mathcal{P}$ a program with no intensional atoms. Then, there is a one-to-one correspondence between the stable models of $\mathcal{P}$ and the partial stable models of the normal logic program $\mathcal{P}$.*

While partial stable models are indeed truth-minimal, this turns out not to be the case for intensional programs due to non-monotonic intensional operators.

Example 7. Consider the operator $|j,k|_a$ representing that an atom is true during all time points in $[j,k]$ for agent a, and not in any interval properly containing $[j,k]$. This operator has the following neighborhood (given W_1 from Ex. 3): $\theta_{|j,k|_a}((i,\ell,t)) = \{W' \subseteq W_1 \mid \{(a,\ell,j),(a,\ell,j{+}1),\ldots,(a,\ell,k)\} \subseteq W'$ and $(a,\ell,j{-}1),(a,\ell,k{+}1) \notin W'\}$. Consider the following program $\mathcal{P}$:

$$[a]_1 \mathtt{resides} \leftarrow \qquad [a]_2 \mathtt{resides} \leftarrow \qquad [a]_3 \mathtt{resides} \leftarrow \sim |1,2|_a \mathtt{resides}$$

For simplicity we restrict ourselves to $W_1^\alpha = \{(i,\alpha,t) \mid i \in A, t \in T\}$. Then this program has two stable models, and one of them is not minimal. Namely, these interpretations are stable: I_1 with $\|\mathtt{resides}\|^{\top}_{I_1} = \|\mathtt{resides}\|^{\mathsf{u}}_{I_1} = \{(a,\alpha,1),(a,\alpha,2)\}$ and I_2 with $\|\mathtt{resides}\|^{\top}_{I_2} = \|\mathtt{resides}\|^{\mathsf{u}}_{I_2} = \{(a,\alpha,1),(a,\alpha,2),(a,\alpha,3)\}$. To see that I_2 is stable, observe first that since $\{(a,\alpha,1),(a,\alpha,2),(a,\alpha,3)\} \notin \theta_{|1,2|_a}(w)$ for any $w \in W_1$, $\||1,2|_a \mathtt{resides}\|^{\top}_{I_2} = \emptyset$, which means that $\mathcal{P}/I_2 = \{[a]_1 \mathtt{resides} \leftarrow ; [a]_2 \mathtt{resides} \leftarrow ; [a]_3 \mathtt{resides} \leftarrow\}$. Clearly, I_2 is the $\sqsubset$-minimal interpretation that satisfies $\mathcal{P}/I_2$. However, $I_1 \sqsubset I_2$ and thus, I_2 is not a truth-minimal stable model.

To counter that, we consider monotonic operators. Formally, given a set of atoms $\mathcal{A}$ and a frame $\mathfrak{F}$, an intensional operator ∇ is said to be *monotonic in* $\mathfrak{F}$ if, for any two interpretations I and I' such that $I \sqsubseteq I'$, we have that $\|\nabla p\|_I^{\dagger} \subseteq \|\nabla p\|_{I'}^{\dagger}$ for every $p \in \mathcal{A}$ and $\dagger \in \{\top, u\}$.

If all intensional operators in a frame are monotonic, then truth-minimality of stable models is guaranteed.

Proposition 2. *Let $\mathcal{A}$ be set of atoms, and $\mathfrak{F}$ a frame in which all intensional operators are monotonic. If I is a stable model of $\mathcal{P}$, then there is no stable model I' of $\mathcal{P}$ such that $I' \sqsubset I$.*

Regarding support, recall that the stable models semantics of normal logic programs satisfies the support property, in the sense that for every atom of a stable model there is a rule that justifies it. In other words, if we remove an atom p from a stable model some rule becomes false in the resulting model. Such rule can be seen as a justification for p being true at the stable model. In the case of intensional logic programs we say that an interpretation $I = \langle W, N, V\rangle$ is *supported* for a program $\mathcal{P}$ if, for every $p \in \mathcal{A}$ and $w \in W$, if $w \in \|p\|^{\top}$, then there is a rule $r \in \mathcal{P}/I_w$ that is not satisfied by I' at w, where $I' = \langle W, N, V'\rangle$ is such that $V'(q) = V(q)$ for $q \neq p$, and $V'(p) = \langle W^{\top} \setminus \{w\}, W^u\rangle$ where $V(p) = \langle W^{\top}, W^u\rangle$.

This notion of supportedness is desirable for intensional logic programs since we also want a justification why each atom is true at each world in a stable model. The following results show that this is indeed the case.

Proposition 3. *Let $\mathcal{A}$ be set of atoms, and $\mathfrak{F}$ a frame. Then, every stable model of a program $\mathcal{P}$ is supported.*

Yet, existence and uniqueness of stable models of a program are not guaranteed, not even for positive programs under the restriction of all operators being monotonic.

Example 8. Let $\mathcal{O} = \{\oplus\}$, $\mathcal{A} = \{p\}$ and $\mathfrak{F} = \langle\{1, 2\}, \{\theta_{\oplus}\}\rangle$ where $\theta_{\oplus}(1) = \theta_{\oplus}(2) = \{\{1\}, \{2\}\}$. Let $\mathcal{P} = \{\oplus p \leftarrow\}$. This program has two stable models: I_1 with $V_1(p) = (\{1\}, \{1\})$ and I_2 with $V_2(p) = (\{2\}, \{2\})$.

The existence of two stable models of the above positive program is caused by the non-determinism introduced by the intensional operator in the head of the rule. Formally, an operator θ of a frame $\mathfrak{F} = \langle W, N\rangle$ is *deterministic* if $\bigcap \theta(w) \in \theta(w)$ for every $w \in W$. A program $\mathcal{P}$ is *deterministic in the head* if, for every rule $r \in \mathcal{P}$ of the form (1), if $A = \nabla p$, then θ_{∇} is deterministic.

We can show that every positive program that is deterministic in the head and only considers monotonic operators has a single minimal model.

Proposition 4. *Given a set of atoms $\mathcal{A}$ and a frame $\mathfrak{F}$, if $\mathcal{P}$ is a positive program that is deterministic in the head and every $\nabla \in \mathcal{O}$ is monotonic in $\mathfrak{F}$, then it has a unique stable model.*

Due to this result, in what follows, we focus on monotonic operators and programs that are deterministic in the head, as this is important for several of the results we obtain subsequently.

Remark 2. This does not mean that non-montonic intensional operators cannot be used in our framework. In fact, we can take advantage of the default negation operator $\sim$ to *define* non-monotonic formulas on the basis of monotonic operators and default negation. E.g., consider again the operator $|j,k|$ from Ex. 7. We can use the following rule to define $|j,k|p$ for some atom $p \in \mathcal{A}$: $|j,k|^{\alpha}_{a}p \leftarrow [j,k]^{\ell}[i]p, \sim [a]^{\alpha}_{j-1}p, \sim [a]^{\alpha}_{k+1}p$.

Among the stable models of a program, we can distinguish the *well-founded models* as those that are minimal in terms of the knowledge order.

Definition 9. *Given a set of atoms $\mathcal{A}$ and a frame $\mathfrak{F}$, an interpretation $I = \langle W, N, V \rangle$ is a* well-founded model of a program $\mathcal{P}$ *if it is a stable model of $\mathcal{P}$, and, for every stable model I' of $\mathcal{P}$, it holds that $I \sqsubseteq_k I'$.*

Example 9 (Example 6 continued). Since I_2 is in fact the unique stable model, it is therefore the well-founded model.

Given our assumptions about monotonicity and determinism in the head, we can show that the well-founded model of an intensional program exists and is unique.

Theorem 1. *Given a set of atoms $\mathcal{A}$, and a frame $\mathfrak{F}$, every program $\mathcal{P}$ has a unique well-founded model.*

This is an important result as a unique model can be computed rather than guessed and checked.

4 ALTERNATING FIXPOINT

In this section, we show how the well-founded model can be efficiently computed. Essentially, we extend the idea of the alternating fixpoint developed for logic programs [21], that builds on computing, in an alternating manner, underestimates of what is necessarily true, and overestimates of what is not false, with the mechanisms to handle intensional inferences.[7]

First, since different pieces of knowledge are inferable in different worlds, we need a way to distinguish between these. Therefore, we introduce labels referring to worlds and apply them to formulas of a given language as well as programs, resulting in formulas $w : A$ and program rules $w : r$ constituting a labelled language $\mathcal{L}_W$ and a labelled program $\mathcal{P}_W$, respectively.

Secondly, three operators are defined to ensure that information is extracted correctly from rules and intensional atoms:

- the immediate consequence operator $T_{\mathcal{P}_W} : \wp(\mathcal{L}_W) \to \wp(\mathcal{L}_W)$ which allows to derive labelled programs atoms occuring in the head of rules in the labelled program $\mathcal{P}_W$ if the atoms in the body of the rule are in the set we apply the operator to.
- the *intensional extraction operator* $IE_{\mathcal{P}_W}(\Delta)$ which allows, for a labelled intensional atom $w : \nabla A$, to derive the labelled atoms $w' : A$ for $w' \in \bigcap \theta_{\nabla}(w')$ that are required to guarantee the truth of $w : \nabla A$.

[7] Due to space restrictions, we are not able to provide full details and examples of this procedure. However, all definitions and relevant propositions are provided in full detail in the appendix.

– the *intensional consequence operator* $IC_\nabla(\Delta)$ which maps labelled atoms to intensional atoms that are implied by the former, i.e. it maps $w_1 : A, \ldots, w_n : A$ to $w : \nabla A$ if $\{w_1, \ldots, w_n\} \in \theta_\nabla(w)$.

These three operators allow us to define a closure operator for a a labelled positive program $\mathcal{P}_W$ as the least fixpoint of:

$$\bigcup_{\nabla \in \mathcal{O}} IC_\nabla \left(\bigcup_{\nabla \in \mathcal{O}^\mathcal{P}} IE_\nabla(T_{\mathcal{P}_W}) \right).$$

Based on this closure operator, the alternating fixpoint procedure can now be defined in the usual way as: Given a frame $\mathfrak{F} = \langle W, N \rangle$ and a program $\mathcal{P}$, we define:

$$\begin{array}{lll} P^0 = \emptyset & P^{i+1} = Cn(\mathcal{P}_W/N^i) & P^\omega = \bigcup_i P^i \\ N^0 = \mathcal{L}_W & N^{i+1} = Cn(\mathcal{P}_W/P^i) & N^\omega = \bigcap_i N^i \end{array}$$

Given a frame $\mathfrak{F}$, for which any $\nabla \in \mathcal{O}$ is monotonic in $\mathfrak{F}$, the alternating fixpoint construction defined above offers a characterization of the well-founded model for programs that are deterministic in the head. In more detail, given a pair $\langle \Delta, \Theta \rangle$ of sets of $\mathcal{L}_W$-formulas, we define a partial interpretation $I(\langle \Delta, \Theta \rangle) = (W, N, V)$ on the basis of Δ as follows: for every $A \in \mathcal{A}$, $V(A) = (\{w \in W \mid w : A \in \Delta\}, \{w \in W \mid w : A \in \Theta\})$. We can then show this correspondence.

Theorem 2. *Given a frame $\mathfrak{F} = \langle W, N \rangle$, and a program $\mathcal{P}$ s.t. every $\nabla \in \mathcal{O}$ is monotonic in $\mathfrak{F}$ and $\mathcal{P}$ is deterministic in the head, then $I(\langle P^\omega, N^\omega \rangle)$ is the well-founded model of $\mathcal{P}$.*

Note that this procedure can be explored for providing explanations for inferences. It is possible to determine the least i such that a labelled program atom is true in P^i. This then allows us to determine justifications building on the construction of the involved operators. We leave exploring this line of research as future work.

5 COMPUTATIONAL COMPLEXITY

In this section, we study the computational complexity of several of the problems considered. We recall that the problem of satisfiability under neighborhood semantics has been studied for a variety of epistemic structures [40]. Here, we consider the problem of determining models for the two notions we established, stable models and the well-founded model, focussing on the propositional case, [8] and we assume familiarity with standard complexity concepts, including oracles and the polynomial hierarchy.

We first provide a result in the spirit of model-checking for programs $\mathcal{P}$. As we do not impose any semantic properties on the neighborhood frames, determining a model for a frame that can be arbitrarily chosen is not meaningful. Thus, we assume a fixed frame $\mathfrak{F}$, fixing the worlds and the semantics of the intensional operators.[9]

[8] Corresponding results for the data complexity of this problem for programs with variables can then be achieved in the usual way [20].

[9] This also aligns well with related work, e.g., for reasoning with time, such as stream reasoning where a finite timeline is often assumed, and avoids the exponential explosion on the number of worlds for satisfiability for some epistemic structures [40].

Proposition 5. *Given a program $\mathcal{P}$ and an interpretation I, deciding whether I is a stable model of $\mathcal{P}$ is in* coNP*, and in* P *if all operators occurring in $\mathcal{P}$ are monotonic.*

This result is due to the minimization of stable models, i.e., we need to check for satisfaction and verify that there is no other interpretation which is smaller (see Def. 8). This also impacts on the complexity of finding a stable model given a fixed frame.

Theorem 3. *Given a program $\mathcal{P}$, deciding whether there is a stable model of $\mathcal{P}$ is in Σ_2^{P}, and in* NP *if all operators occurring in $\mathcal{P}$ are monotonic.*

Thus, if all operators are monotonic the complexity results do coincide with that of normal logic programs (without intensional atoms) [20], which indicates that monotonic operators do not add an additional burden in terms of computational complexity.

Now, if we in addition consider programs that are deterministic in the head, then we know that there exists the unique well-founded model (see Thm. 1). As we have shown, this model can be computed efficiently (see Thm. 2), and we obtain the following result in terms of computational complexity.

Theorem 4. *Given a program $\mathcal{P}$ that is deterministic in the head and all operators occurring in $\mathcal{P}$ are monotonic, computing the well-founded model of $\mathcal{P}$ is* P*-complete.*

Note that this result is indeed crucial in contexts were reasoning with a variety of intensional concepts needs to be highly efficient.

6 RELATED WORK

In this section, we discuss related work establishing relations to relevant formalisms in the literature.

Intensional logic programs were first defined by Orgun and Wadge [34] focussing on the existence of models in function of the properties of the intensional operators. Only positive programs are considered, and thus our approach covers the previous work. Since [34] covers classical approaches for intensional reasoning, such as TempLog [1] and MoLog [17], our work applies to these as well.

It also relates to more recent work with intensional operators, and we first discuss two prominent approaches in the area of stream reasoning.

LARS [10] assumes a set of atoms $\mathcal{A}$ and a stream $S = (T, v)$, where T is a closed interval of the natural numbers and v is an evaluation function that defines which atoms are true at each time point of T. Several temporal operators are defined, including expressive window operators, and answer streams, a generalization of FLP-semantics, are employed for reasoning. A number of related approaches are covered including CQL [6], C-SPARQL [8], and CQELS [37]. Among the implementations exists LASER [9], which focuses on a considerable fragment, called plain LARS. We can represent a plain LARS program and have shown (in the appendix) that there is a one-to-one correspondence between answer streams of the program and the total stable models of the corresponding intensional logic program. In addition, we can apply our well-founded semantics, since the operators applied in plain LARS are monotonic and deterministic.

Hence, our work also provides a well-founded semantics for plain LARS, i.e., we allow the usage of unrestricted default negation while preserving polynomial reasoning.

ETALIS [5] aims at *complex event processing*. It assumes as input atomic events with a time stamp and uses *complex events*, based on Allen's interval algebra [4], that are associated with a time *interval*, and is therefore considerably different from LARS (which considers time points). It contains no negation in the traditional sense, but allows for a negated pattern among the events. Many of the complex event patterns from ETALIS can be captured as neighborhood functions in our framework. However, ETALIS also makes use of some event patterns that would result in a non-monotonic operator, such as the negated pattern $\mathtt{not}(p)[q, r]$ which expresses that p is not the case in the interval between the end time of q and the starting time of r. We conjecture that such a negation can be modelled with help of the non-monotonic default negation $\sim$ and monotonic operators (see also Remark 2).

Other formalisms that extend logic programming with intensional operators include *Deontic Logic Programs* [25], *Answer Set Programming Modulo Theories extended to the Qualitative Spatial Domain* [42] and *Metric Temporal Answer Set Programming* [15]. In future work, we plan to study instantiations of our general framework that represent (fragments) of these languages.

7 CONCLUSIONS

We have presented intensional logic programs that allow defeasible reasoning with intensional concepts and streams of data, and introduced a novel three-valued semantics based on the neighborhood semantics [35] and partial stable models [38]. We have studied the characteristics of our semantics for monotonic intensional operators and programs that only admit deterministic operators in the heads of the rules, and shown that a unique minimal model, the well-founded model, exists and can be computed in polynomial time. Still, several relevant approaches in the literature can be covered, and for one of them our work also provides a well-founded semantics for the first time.

In terms of future work, several generalizations are possible, for example, allowing for first-order formulas in the programs and non-deterministic intensional operators. We can possibly resort to techniques from well-founded semantics for disjunctive logic programs [32] to resolve the non-determinism that occurs when studying the latter. Finally, the integration with taxonomic knowledge in the form of description logic ontologies [7] may also be worth pursuing as applications sometimes require both (see e.g. [2,3,29]). Hybrid MKNF knowledge bases [33] are a more prominent approach among the existing approaches for combining non-monotonic rules and such ontologies, and the well-founded semantics for these [31] together with its efficient implementation [30] may prove fruitful for such an endeavour.

Acknowledgements The authors are indebted to the anonymous reviewers of this paper for helpful feedback. The authors were partially supported by FCT project RIVER (PTDC/CCI-COM/30952/2017) and by FCT project NOVA LINCS (UIDB/04516/2020). J. Heyninck was also supported by the German National Science Foundation under the DFG-project CAR (Conditional Argumentative Reasoning) KE-1413/11-1.

References

1. Abadi, M., Manna, Z.: Temporal logic programming. J. Symb. Comput. **8**(3), 277–295 (1989)
2. Alberti, M., Gomes, A.S., Gonçalves, R., Leite, J., Slota, M.: Normative systems represented as hybrid knowledge bases. In: CLIMA. LNCS, vol. 6814, pp. 330–346. Springer (2011)
3. Alberti, M., Knorr, M., Gomes, A.S., Leite, J., Gonçalves, R., Slota, M.: Normative systems require hybrid knowledge bases. In: AAMAS. pp. 1425–1426. IFAAMAS (2012)
4. Allen, J.F.: Maintaining knowledge about temporal intervals. In: Readings in qualitative reasoning about physical systems, pp. 361–372. Elsevier (1990)
5. Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Stream reasoning and complex event processing in ETALIS. Semantic Web **3**(4), 397–407 (2012)
6. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. VLDB J. **15**(2), 121–142 (2006)
7. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
8. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: C-SPARQL: a continuous query language for RDF data streams. Int. J. Semantic Computing **4**(1), 3–25 (2010)
9. Bazoobandi, H.R., Beck, H., Urbani, J.: Expressive stream reasoning with LASER. In: Procs. of ISWC. LNCS, vol. 10587, pp. 87–103. Springer (2017)
10. Beck, H., Dao-Tran, M., Eiter, T.: LARS: A logic-based framework for analytic reasoning over streams. Artif. Intell. **261**, 16–70 (2018)
11. Beirlaen, M., Heyninck, J., Straßer, C.: Structured argumentation with prioritized conditional obligations and permissions. Journal of Logic and Computation **29**(2), 187–214 (2019)
12. Brandt, S., Kalayci, E.G., Ryzhikov, V., Xiao, G., Zakharyaschev, M.: Querying log data with metric temporal logic. J. Artif. Intell. Res. **62**, 829–877 (2018)
13. Brenton, C., Faber, W., Batsakis, S.: Answer set programming for qualitative spatio-temporal reasoning: Methods and experiments. In: Technical Communications of ICLP. OASICS, vol. 52, pp. 4:1–4:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
14. Brewka, G., Ellmauthaler, S., Gonçalves, R., Knorr, M., Leite, J., Pührer, J.: Reactive multi-context systems: Heterogeneous reasoning in dynamic environments. Artif. Intell. **256**, 68–104 (2018)
15. Cabalar, P., Dieguez, M., Schaub, T., Schuhmann, A.: Towards metric temporal answer set programming. Theory and Practice of Logic Programming **20**(5), 783–798 (2020)
16. Caminada, M., Sá, S., Alcântara, J., Dvořák, W.: On the equivalence between logic programming semantics and argumentation semantics. International Journal of Approximate Reasoning **58**, 87–111 (2015)
17. del Cerro, L.F.: MOLOG: A system that extends PROLOG with modal logic. New Generation Comput. **4**(1), 35–50 (1986)
18. Chellas, B.F.: Modal Logic: An Introduction. Cambridge University Press (1980)
19. Chen, Y., Wan, H., Zhang, Y., Zhou, Y.: dl2asp: implementing default logic via answer set programming. In: European Workshop on Logics in Artificial Intelligence. pp. 104–116. Springer (2010)
20. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Comput. Surv. **33**(3), 374–425 (2001)
21. Gelder, A.V.: The alternating fixpoint of logic programs with negation. In: Procs. of SIGACT-SIGMOD-SIGART. pp. 1–10. ACM Press (1989)
22. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM **38**(3), 620–650 (1991)

23. Gelfond, M.: Answer sets. In: Handbook of Knowledge Representation, Foundations of Artificial Intelligence, vol. 3, pp. 285–316. Elsevier (2008)
24. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Comput. **9**(3-4), 365–385 (1991)
25. Gonçalves, R., Alferes, J.J.: Specifying and reasoning about normative systems in deontic logic programming. In: Procs. of AAMAS. pp. 1423–1424. IFAAMAS (2012)
26. Gonçalves, R., Knorr, M., Leite, J.: Evolving multi-context systems. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 263, pp. 375–380. IOS Press (2014)
27. Governatori, G., Rotolo, A., Riveret, R.: A deontic argumentation framework based on deontic defeasible logic. In: International Conference on Principles and Practice of Multi-Agent Systems. pp. 484–492. Springer (2018)
28. Izmirlioglu, Y., Erdem, E.: Qualitative reasoning about cardinal directions using answer set programming. In: Procs. of AAAI. pp. 1880–1887. AAAI Press (2018)
29. Kasalica, V., Gerochristos, I., Alferes, J.J., Gomes, A.S., Knorr, M., Leite, J.: Telco network inventory validation with nohr. In: LPNMR. LNCS, vol. 11481, pp. 18–31. Springer (2019)
30. Kasalica, V., Knorr, M., Leite, J., Lopes, C.: NoHR: An overview. Künstl Intell (2020)
31. Knorr, M., Alferes, J.J., Hitzler, P.: Local closed world reasoning with description logics under the well-founded semantics. Artif. Intell. **175**(9-10), 1528–1554 (2011)
32. Knorr, M., Hitzler, P.: A comparison of disjunctive well-founded semantics. In: FAInt. CEUR Workshop Proceedings, vol. 277. CEUR-WS.org (2007)
33. Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM **57**(5), 30:1–30:62 (2010)
34. Orgun, M.A., Wadge, W.W.: Towards a unified theory of intensional logic programming. The Journal of Logic Programming **13**(4), 413–440 (1992)
35. Pacuit, E.: Neighborhood semantics for modal logic. Springer (2017)
36. Panagiotidi, S., Nieves, J.C., Vázquez-Salceda, J.: A framework to model norm dynamics in answer set programming. In: MALLOW (2009)
37. Phuoc, D.L., Dao-Tran, M., Parreira, J.X., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Procs. of ISWC. LNCS, vol. 7031, pp. 370–388. Springer (2011)
38. Przymusinski, T.C.: Stable semantics for disjunctive programs. New Generation Comput. **9**(3/4), 401–424 (1991)
39. Suchan, J., Bhatt, M., Walega, P.A., Schultz, C.P.L.: Visual explanation by high-level abduction: On answer-set programming driven reasoning about moving objects. In: Procs. of AAAI. pp. 1965–1972. AAAI Press (2018)
40. Vardi, M.Y.: On the complexity of epistemic reasoning. In: Procs. of LICS. pp. 243–252. IEEE Computer Society (1989)
41. Walega, P.A., Kaminski, M., Grau, B.C.: Reasoning over streaming data in metric temporal datalog. In: Procs. of AAAI. pp. 3092–3099. AAAI Press (2019)
42. Walega, P.A., Schultz, C.P.L., Bhatt, M.: Non-monotonic spatial reasoning with answer set programming modulo theories. TPLP **17**(2), 205–225 (2017)