Adapting Approximation Fixpoint Theory to Nondeterministic Hybrid Reasoning

Spencer Killen[1], Jia-Huai You[1]

[1]University of Alberta, Edmonton, Canada

Abstract

Approximation fixpoint theory (AFT) is an abstract, algebraic framework for the study of operators and their fixpoints on bilattices. It is based on *approximating operators* (*approximators* for short) and has seen application in expressing semantics for a variety of nonmonotonic languages. The original AFT only treats consistent, symmetric, and exact approximators. Consequently, it lacks the means to deal with inconsistencies which may arise naturally in systems of hybrid reasoning. This problem was addressed by a generalization of AFT by Liu and You [1], which alters the exactness condition so that inconsistencies can be supported. However, neither the original AFT nor this generalized AFT is capable of expressing nondeterministic semantics, which is essential for characterizing disjunctive knowledge. The recent proposal of nondeterministic AFT by Heyninck et al. [2] only deals with consistent and exact approximations, and its definitions of orderings, fixpoints, and monotonicity deviate from that of the original AFT. Due to differences in these core definitions, it is highly challenging, if not impossible, to generalize nondeterministic AFT to handle inconsistencies. In this paper, we present a new framework which relies on a family of approximators in Liu and You's formulation to express nondeterministic semantics abstractly and algebraically. As an application, we show how to capture the partial stable semantics of (disjunctive) hybrid MKNF knowledge bases.

Keywords

approximation fixpoint theory, hybrid MKNF, disjunctive ASP

1. Introduction

Approximation Fixpoint Theory (AFT) [3] is a powerful framework that leverages pairs of elements (approximations) to construct operators (approximators) that can capture the fixpoints of nonmonotonic operators defined on bilattices and provide characterizations of semantics for nonmonotonic systems. A bilattice is a partially ordered set consisting of a set of pairs (x, y), where x and y are elements of a complete lattice and an ordering. In the ideal situation where the value of x is less or equal to y in the underlying ordering, (x, y) represents an interval where x is a lower bound and y an upper bound. In the context of logic, x is often used to represent the set of true atoms and y the set of possibly true atoms (so that any atom not in y is false and atoms that are neither true nor false are *undefined*). An *exact* pair (x, x) can be viewed as a two-valued interpretation.

In AFT, we are given the tools of algebra, in terms of operators and the fixpoints built around them, to express stable semantics as well as partial stable semantics, and the main focus while characterizing a semantics is to define an appropriate approximator. Note that in the context of logic programming, partial semantics is often closely related to the simplification of a program during grounding and to constraint propagation in search; namely, in these processes, we are interested in which atoms we can infer to be true or false towards the goal of constructing two-valued stable solutions. AFT's generality and ease of use has made it a popular approach for a variety of applications (e.g. [4, 5, 6, 7, 8, 9]).

However, AFT requires operators to be deterministic and *exact*. It is incapable of expressing nondeterministic semantics and dealing with inconsistencies in general, which are required for capturing disjunctive knowledge and dealing with classical hybrid reasoning, respectively. An exact approximator must map an exact pair to an exact pair. In the context of logic programming, this is to say that, given a two-valued interpretation, a logic program must map it to a two-valued interpretation. This requirement is too strict for hybrid reasoning, where different reasoning components together may lead to a conflict,

ASPOCP 2024: 17th Workshop on Answer Set Programming and Other Computing Paradigms, October, 2024, Dallas, USA.

which shall be represented by a pair (x, y) where x is not a "lower value" than that of y. In general, a hybrid knowledge base may even map a partial interpretation to an inconsistent one.

This paper shows how to adapt AFT to capture nondeterministic semantics. This results in an extension of the current development of AFT. That is, we propose a new abstract, algebraic framework to represent and characterize nondeterministic semantics relying only on the notations and definitions of the current AFT. To accommodate inconsistencies, we lift the notion of conditional stable fixpoints from Liu and You [1]'s generalized AFT to a nondeterministic setting. As a demonstration, we instantiate our extension to apply it to hybrid MKNF (Minimal Knowledge and Negation as Failure) [10].

The unique challenges of capturing disjunctive hybrid MKNF are twofold: inconsistencies may arise from the ontology and disjunctive rules and require nondeterministic treatment. While AFT has recently been extended to nondeterministic operators [2], this framework is ill-suited for inconsistencies: Heyninck et al.'s Definition 15 and Proposition 5 [2] demonstrate how inconsistent pairs are excluded in their framework. Bi et al. [11] construct a generalized variant of AFT that relaxes AFT to deal with inconsistent approximations and in prior work [12], we applied this theory to lift this framework to disjunctive hybrid MKNF. However, this application is separate from AFT and thus could not easily be applied elsewhere. Here, we recharacterize the semantics of disjunctive hybrid MKNF knowledge bases using a set of approximators. We provide a method of incorporating nondeterminism and inconsistencies into AFT.

This paper is outlined as follows. Section 2 introduces prelimaries for lattice theory (Section 2.1), and approximation fixpoint theory (Section 2.2). Next, we lift AFT to the nondeterministic setting (Section 3). In Section 4, we apply our theory to hybrid MKNF knowledge bases after we introduce prelimaries for disjunctive logic programs (Section 4.1) and hybrid MKNF knowledge bases (Section 4.2). Finally, we conclude and discuss future work in Section 5.

2. Preliminaries

2.1. Lattice Theory

We review common lattice theory [13] to establish the notation and terminology used throughout this work. A *preorder* $\langle S, \preceq \rangle$ is a relation $\preceq$ over a non-empty set of elements S that satisfies: *reflexivity* ($\forall x \in S, x \preceq x$) and *transitivity* ($\forall x, y, z \in S$ if $x \preceq y$ and $y \preceq z$ then $x \preceq z$). If the order $\preceq$ also satisfies *antisymmetry*, ($\forall x, y \in S,$ if $x \preceq y$ and $y \preceq x$ then $x = y$), then we call it a *partial order* (or *poset* for short). Without confusion, we sometimes refer to a poset $\langle S, \preceq \rangle$ simply by S or $\preceq$. Given a preorder $\langle \preceq, S \rangle$, we call an element $x \in S$ an *upper bound* (resp. a *lower bound*) of a subset $Q \subseteq S$ if $\forall y \in Q, y \preceq x$ (resp. $\forall y \in Q, x \preceq y$). An upper bound of Q w.r.t. a poset $\langle S, \preceq \rangle$ is a *least upper bound*, denoted as $\bigvee_{\preceq}(Q)$ (resp. *greatest lower bound*, denoted as $\bigwedge_{\preceq}(Q)$) if it is a lower bound of the set of all upper bounds of Q (resp. an upper bound of the set of all lower bounds of Q). A *complete lattice* $\langle \mathcal{L}, \preceq \rangle$ is a partial order s.t. a least upper bound and greatest lower bound exists for every subset $S \subseteq \mathcal{L}$. For a complete lattice $\langle \mathcal{L}, \preceq \rangle$, we denote $\bigwedge_{\preceq}(\mathcal{L})$ as $\bot_{\preceq}$ and $\bigvee_{\preceq}(\mathcal{L})$ as $\top_{\preceq}$ when $\mathcal{L}$ is clear from context or simply as $\bot$ and $\top$ when the lattice's relation is unambiguous. Given a complete lattice $\langle \mathcal{L}, \preceq \rangle$, we say a function $f : \mathcal{L} \to \mathcal{L}$ is *monotone* if for each $x, y \in \mathcal{L}$, having $x \preceq y$ implies $f(x) \preceq f(y)$.

Every preorder $\langle \mathcal{L}_a, \preceq \rangle$ has a dual ordering $\langle \mathcal{L}_a, \succeq \rangle$ where the ordering has been "flipped". That is, $(a \preceq b) \iff (b \succeq a)$. We also define a strict variant $\prec$ to mean two elements are comparable and not equal, i.e., $(a \prec b) \iff ((a \preceq b) \land a \neq b)$. We use $\mathcal{L}_1 \times \mathcal{L}_2$ to denote the cartesian product between two sets and $\mathcal{L}^2$ to mean $\mathcal{L} \times \mathcal{L}$. We use $\wp(S)$ to indicate the set of all subsets of S and use $\wp^o(S)$ to denote the set of all non-empty subsets of S.

We use subscript notation to denote the projection of components of a tuple $s \in \mathcal{L}^2$, that is, $(s_1, s_2) = s$.

Given a complete lattice $\langle \mathcal{L}, \preceq \rangle$ and a $\preceq$-monotone function $o : \mathcal{L} \to \mathcal{L}$, an element $x \in \mathcal{L}$ is a *fixpoint* of o if $x = o(x)$. The set of all fixpoints of o has a lower bound that is also a fixpoint of o [14]. We call this element the *least fixpoint* of o and denote it as $\mathbf{lfp}_{\preceq_\alpha} o$.

We denote partially applied functions using a "·" in place of arguments to be filled in, that is, for a function $o(T, P) : \mathcal{L}^2 \to \mathcal{L}^2$, we write $o(\cdot, P)$ (resp. $o(T, \cdot)$) to mean $\lambda x \Rightarrow \ o(x, P)$ (resp. $\lambda x \Rightarrow o(T, x)$). If a "·" is used within a tuple projection, the body of the lambda abstraction includes the projections, for example,

$$f(x, \cdot)_1 = \big(\lambda y \Rightarrow (f(x, y)_1)\big) \qquad\qquad (\textit{where } f(x, y) : \mathcal{L}^2 \to \mathcal{L}^2)$$

This makes it possible to write $\mathbf{lfp}_{\preceq} o(T, \cdot)_1$ to express the $\preceq$-least fixpoint of o as partially applied with T. We use $\boldsymbol{min}_{\preceq}(S)$ to denote the set $\boldsymbol{min}_{\preceq}(S) := \{b \in S \mid \neg\exists b' \in S, b' \prec b\}$.

2.2. Generalized AFT

Introduced by Denecker et al. [2000], AFT (Approximation Fixpoint Theory) is a framework for defining operators over a bilattice. The framework captures a variety of nonmonotonic semantics through fixpoints of a stable revision operator. Bi et al. [11] propose a generalization of AFT (Approximation Fixpoint Theory) that relaxes the exact condition on approximators. That is, in traditional AFT, an operator must map exact pairs to exact pairs. In generalized AFT, an operator can also map exact pairs to inconsistent pairs. This enhancement enables the creation of more precise approximators for systems that blend classical and nonmonotonic reasoning.

First, we define the bilattice on which both original [3] and generalized [11] AFT rely. Given a complete lattice $\langle \mathcal{L}, \preceq \rangle$, for any two pairs $(x', y'), (x, y) \in \mathcal{L}^2$,

$$(x', y') \preceq_t^2 (x, y) \iff ((x' \preceq x) \land (y' \preceq y))$$
$$(x', y') \preceq_p^2 (x, y) \iff ((x' \preceq x) \land (y' \succeq y))$$

We introduce the generalization of AFT from Bi et al. [11] and Liu and You [1].

Definition 2.1. *A* (generalized) approximator $o : \mathcal{L}^2 \to \mathcal{L}^2$ *over a complete lattice* $\langle \mathcal{L}, \preceq \rangle$ *is a* $\preceq_p^2$*-monotone function such that for any element* $x \in \mathcal{L}$ *if*

$$o(x, x)_1 \preceq o(x, x)_2 \textit{ then } o(x, x) = (o(x, x)_1, o(x, x)_1)$$

Traditional AFT [3] requires approximators to map *exact* pairs (a pair (x, x)) to exact pairs. Here, we only require that exact pairs are not mapped to inexact, *consistent* pairs (a pair (x, y) is *consistent* if $x \preceq y$). For example, the following approximator is possible in this generalized AFT [11], but not in traditional AFT [3].

Example 1. *Given a complete lattice* $\{\bot, \top\}$*, define* $o : \mathcal{L}^2 \to \mathcal{L}^2$ *as follows.*

$$o(x, y) := \begin{cases} (\top, \bot) & \textit{if } x = y = \bot \\ (x, y) & \textit{otherwise} \end{cases}$$

The above operator is $\preceq_p^2$*-monotone as* $o(\bot, \bot) = o(\top, \bot)$*. Here, the exact pairs* $(\bot, \bot)$ *and* $(\top, \top)$ *map to an inconsistent and exact pair respectively. Thus,* o *is an approximator.*

We can reuse the original AFT [3] definition of stable fixpoints for approximators [1], which are shown to be well-defined; however, these fixpoints might not be consistent. As a result, some stable fixpoints of approximators do not correspond to intended models.

Definition 2.2. *Given an approximator* o*, the* stable revision operator $S(o) : \mathcal{L}^2 \to \mathcal{L}^2$ *is defined as follows:* $S(o)(x, y) := (\mathbf{lfp}_{\preceq_\alpha}(o(\cdot, y)_1), \mathbf{lfp}_{\preceq_\alpha}(o(x, \cdot)_2))$.

For an approximator o, we refer to the fixpoints of S(o) as *stable fixpoints*. The smallest fixpoint w.r.t. $\preceq_p^2$ is called the *least stable fixpoint*. Critically, this modified variant of AFT maintains the truth minimality property of traditional AFT.

Proposition 2.1. *Let (x, y) be a stable fixpoint of a approximator o over $\langle \mathcal{L}, \preceq_p^2 \rangle$.*

- *(x, y) is a fixpoint of o, and*
- *there does not exist a fixpoint (x', y') (distinct from (x, y)) of o s.t. $(x', y') \prec_t^2 (x, y)$*

For the above, we can follow Denecker et al. [3]'s proof (of Theorem 4), which does not rely on exactness.

3. Generalized Approximator Sets

In this section, we abstract and generalize the application of AFT on disjunctive hybrid MKNF [12] so that it works within the confines of AFT and can be applied to other semantics. Namely, we obtain a general method of lifting deterministic stable revision to nondeterministic semantics. Optionally, in this framework one can also choose to eliminate select stable fixpoints, namely the stable fixpoints that correspond to inconsistencies.

When they apply their AFT to hybrid MKNF knowledge bases, Liu and You [1] couple their approximator with a condition that checks whether the stable fixpoint corresponds to an MKNF model. This is a critical step in addressing inconsistencies because it allows us to prevent inconsistent stable fixpoints. As we will see later on, additional complications w.r.t. $\preceq_t^2$-minimality may arise when lifting this theory to support nondeterministic semantics.

We lift approximators to a nondeterministic setting by considering sets of approximators coupled with a condition to filter unintended stable fixpoints.

Definition 3.1. *A (generalized)* a-set *$\langle H, \Theta \rangle$ is a set of approximators H over the complete lattice $\langle \mathcal{L}^2, \preceq_p^2 \rangle$ paired with its* acceptance relation *$\Theta \subseteq (H \times \mathcal{L}^2)$.*

By harnessing a family of deterministic approximators, we can characterize nondeterministic semantics. A generalized approximator can have unintended stable fixpoints, that is, stable fixpoints that do not correspond to the intended semantics. The acceptance relation enables us to filter out unintended stable fixpoints. However, use of the relation is optional and the theory can be applied without it by fixing the acceptance relation to be $H \times \mathcal{L}^2$ (the maximal acceptance relation).

We lift the stable revision operator to a-sets.

Definition 3.2. *A pair $(x, y) \in \mathcal{L}$ s.t. $x \preceq y$ is a* (generalized) a-stable fixpoint *of an a-set $\langle H, \Theta \rangle$ if*

$$(x, y) \in \boldsymbol{min}_{\preceq_t^2} \{ S(h)(x, y) \mid (h, (x, y)) \in \Theta \}$$

The mechanism above applies stable revision to all approximators in the a-set that are part of the acceptance relation. Each image has an opportunity to block a pair from being an a-set by computing a $\prec_t^2$-smaller pair. In their notion of stable revision, Liu and You [1] apply a condition to filter some stable fixpoints, which we generalize here as the "acceptance relation". What's new here is that the condition also plays a role in determinining whether a computed pair can "block" another pair from being an a-stable fixpoint. By using the maximal acceptance relation, the condition does not remove any stable fixpoints (and the definitions can be further simplified). For any a-set stable fixpoint (x, y) that is a stable fixpoint of $h \in H$, we can remove $(h, (x, y))$ from Θ to prevent (x, y) from being an a-stable fixpoint. Additionally, if a stable fixpoint (x, y) of $h \in H$ is not an a-stable fixpoint due to an approximator $h' \in H$ s.t. $S(h')(x, y) \prec_t^2 (x, y)$, we can remove $(h', (x, y))$ from Θ in effort to make $(h, (x, y))$ an a-stable fixpoint of $\langle H, \Theta \rangle$. This will remove h' from participating in the consideration of (x, y) as an a-stable fixpoint. These points are demonstrated concretely in the coming Example 2.

The property of $\preceq_t^2$-minimality is critical to stable fixpoints. Given any two stable fixpoints (x', y') and (x, y) of h, we have that $(x', y') \prec_t^2 (x, y)$ does not hold. By using $\boldsymbol{min}_{\preceq_t^2}$ on the stable fixpoint of approximators in H, we ensure this property also holds for a-stable fixpoints. First, we demonstrate this property without using the acceptance relation (i.e., we fix the relation to be maximal).

Proposition 3.1. *For an a-set $\langle H, (H \times \mathcal{L}^2)\rangle$ and the set W, consisting of its a-stable fixpoints, we have $W = \boldsymbol{min}_{\preceq_t^2}(W)$.*

Introducing the acceptance relation, i.e., using a smaller relation than in the above, introduces some complications w.r.t. the minimality of stable fixpoints. The acceptance relation can add or remove a-stable fixpoints. Thus, truth-minimality may be disrupted. We generalize the above proposition to handle any acceptance relation by imposing a condition on this relation.

Proposition 3.2. *For an a-set $\langle H, \Theta\rangle$ and the set W consisting of its a-stable fixpoints, we have $W = \boldsymbol{min}_{\preceq_t^2}(W)$ if Φ satisfies the following condition,*

$$\forall(h', (x', y'))(h, (x, y)) \in \Theta^* \text{ where } (x', y') \prec_t^2 (x, y) \text{ we have}$$
$$\exists(h'', (x'', y'')) \in \Theta^*, \text{ s.t. } (x'', y'') \prec_t^2 (x, y), (h'', (x, y)) \in \Theta$$

where Θ^* denotes the subset of Θ consisting of every pair $(h, (x, y))$ s.t. $S(h)(x, y) = (x, y)$.

Proof. Assume the condition holds for $\langle H, \Theta\rangle$ and let (x', y') and (x, y) be two stable fixpoints of $h' \in H$ and $h \in H$ respectively s.t. $(x', y') \preceq_t^2 (x, y)$ and both pairs are a-stable fixpoints of $\langle H, \Theta\rangle$. We intend to show $(x', y') = (x, y)$ and it will follow that $W = \boldsymbol{min}_{\preceq_t^2}(W)$. Assume for the sake of contradiction, $(x', y') \neq (x, y)$. It follows from our initial assumption of $(x', y') \preceq_t^2 (x, y)$ that $(x', y') \prec_t^2 (x, y)$. By the condition, we have a stable fixpoint (x'', y'') of $h'' \in H$ s.t. $(x'', y'') \prec_t^2 (x, y)$. We briefly follow Denecker et al. [3]'s proof (Theorem 4). We have $x'' \preceq x$ and $y'' \preceq y$. The $\preceq_p^2$-monotonicity of h'' ensures that $h''(\cdot, y')_2$ is $\preceq$-antimonotone, that is, $h''(x, y'')_2 \preceq h''(x'', y'')_2 = y''$. Thus, y'' is a prefixpoint of $h''(x, \cdot)$ and the least fixpoint of $h''(x, \cdot)$ is less than y'' [14]. We have $\mathbf{lfp}_{\preceq}\, h''(x, \cdot) \preceq y''$, with $y'' \preceq y$, it follows by transitivity that $(\mathbf{lfp}_{\preceq}\, h''(x, \cdot)) = S(h'')(x, y)_2 \preceq y$. We can apply a similar approach to show $S(h'')(x, y)_1 \preceq x$, thus $S(h'')(x, y) \preceq_t^2 (x, y)$. Because (x, y) is an a-stable fixpoint of $\langle H, \Theta\rangle$, we have $(x, y) \in \boldsymbol{min}_{\preceq_t^2}\{S(h)(x, y) \mid (h, (x, y)) \in \Theta\}$. With $(h'', (x, y)) \in \Theta$ and $S(h)(x, y) = (x, y)$, it follows that $\neg(S(h'')(x, y) \prec_t^2 (x, y))$. With this and $S(h'')(x, y) \preceq_t^2 (x, y)$, it follows that $S(h'')(x, y) = (x, y)$, that is, (x, y) is a stable fixpoint of h''. With two stable fixpoints $(x'', y'') \prec_t^2 (x, y)$ of h'', we contradict Proposition 2.1.

□

Intuitively, the condition in the proposition above ensures that smaller (w.r.t. $\preceq_t^2$) stable fixpoints must block larger stable fixpoints. We demonstrate this concretely in the following.

Example 2. *Let $\langle \mathcal{L}, \preceq\rangle$ be a complete lattice s.t. $\mathcal{L} = \{\bot, \top\}$ and $\bot \preceq \top$. Let $h_\bot$ and $h_\top$ be two constant functions over $\mathcal{L}^2$ mapping to $(\bot, \bot)$ and $(\top, \top)$ respectively. With $H = \{h_\bot, h_\top\}$, let $\Theta_1 \coloneqq H \times \mathcal{L}^2$. Clearly, $(\bot, \bot)$ and $(\top, \top)$ are stable fixpoints of $h_\bot$ and $h_\top$ respectively. While $(\bot, \bot)$ is an a-stable fixpoint of $\langle H, \Theta_1\rangle$, $(\top, \top)$ is not because $(h_\bot, (\top, \top)) \in \Theta_1$ and $S(h_\bot)(\top, \top) \prec_t^2 (\top, \top)$.*

For $(\top, \top)$ to be an a-stable fixpoint, we must remove $(h_\bot, (\top, \top)) \in \Theta_1$. but this would violate the condition on the acceptance relation in Proposition 3.2. That is, with $(\bot, \bot) \prec_t^2 (\top, \top)$ and $(h_\bot, (\bot, \bot)), (h_\top, (\top, \top)) \in \Theta$. Without this condition, a-stable fixpoints would not be $\preceq_t^2$-minimal.

Let $\Theta_2 \coloneqq \{h_\top\} \times \mathcal{L}^2$. We have that $(\top, \top)$ is a a-stable fixpoint of $\langle H, \Theta_2\rangle$, but $(\bot, \bot)$ is not. Thus, altering the acceptance relation can both add and remove a-stable fixpoints.

A-sets and a-stable fixpoints generalize our prior methods [12] to capture the semantics of disjunctive hybrid MKNF using a set of operators. Now that we've expressed this theory at the algebra level, new opportunities for its application are unlocked.

4. Application to Hybrid MKNF Knowledge Bases

In this section, we apply the theory from the previous section to disjunctive hybrid MKNF knowledge bases. This demonstrates that nothing was lost in generalizing the previous theory that used sets of operators (not approximators) to capture the semantics [12]. First, we cover the necessary preliminaries.

4.1. Preliminaries: Disjunctive Logic Programs

Disjunctive stable semantics extends answer set programming by allowing rules with disjunctions in their heads [15]. This feature raises the expressivity class of programs by enabling them to capture problems in a higher level of the polynomial hierarchy [16]. Furthermore, disjunctive rules allow succinct representation of nondeterministic solutions to problems.

The partial stable model semantics extend stable model semantics by adding an additional truth value (undefined), which is to be assigned to atoms with an unknown truth value. We introduce Przymusinski [17]'s three-valued semantics for disjunctive logical programs. A *disjunctive logic program* (DLP) $\mathcal{P}$ is a set of rules, objects with three components: a head, a positive body, and a negative body. We denote these components as *head*(r), *body*$^+(r)$, and *body*$^-(r)$ respectively, and each is a set of ground atoms from a propositional language $Atoms(\mathcal{P})$. We write a rule r as follows.

$$h_1,\ \ldots,\ h_i \leftarrow p_1,\ \ldots,\ p_j,\ \mathbf{not}\, n_1,\ \ldots,\ \mathbf{not}\, n_k$$

where *head*$(r) = \{h_1, \ldots,\ h_i\}$, *body*$^+(r) = \{p_1, \ldots,\ p_j\}$, and *body*$^-(r) = \{n_1, \ldots,\ n_k\}$. We use *body*$(r)$ to denote the entire expression to the right of the arrow above (maintaining the **not** syntax). A rule r is *normal* if its head contains exactly one atom (i.e. $|$*head*$(r)| = 1$), and a program $\mathcal{P}$ is *normal* if it consists solely of normal rules.

A (three-valued) *interpretation* (T, P) of a disjunctive logic program $\mathcal{P}$ is a pair of sets containing atoms from $Atoms(\mathcal{P})$ s.t. $T \subseteq P \subseteq Atoms(\mathcal{P})$. An interpretation (T, P) assigns each atom $a \in Atoms(\mathcal{P})$ to one of three truth values: $\mathbf{t}rue$ (iff $a \in T \wedge a \in P$), **u***ndefined* (iff $a \notin T \wedge a \in P$), and $\mathbf{f}alse$ (iff $a \notin T \wedge a \notin P$). This logic leverages two orderings between interpretations, a truth ordering $\preceq_t^2$ and a precision ordering $\preceq_p^2$. Both ordering are obtained from the bilattice on $\langle \wp(Atoms(\mathcal{P})), \subseteq\rangle$. Note that $\wp(Atoms(\mathcal{P}))$ contains inconsistent pairs, i.e., not every pair is an interpretation. The $\preceq_t^2$ ordering respects the linear ordering $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$ while the latter uses the partial ordering composed of $\mathbf{u} \leq \mathbf{f}$ and $\mathbf{u} \leq \mathbf{t}$ where $\mathbf{f}$ and $\mathbf{t}$ are incomparable. Intuitively, a rule is satisfied if some atom in its head is at least as true (w.r.t. $\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$) as the conjunction of its body atoms. Negation changes $\mathbf{f}$ to $\mathbf{t}$ and $\mathbf{t}$ to $\mathbf{f}$ while mapping $\mathbf{u}$ to itself. We use the bilattices [18] that lift the truth-ordering ($\mathbf{f} \leq \mathbf{u} \leq \mathbf{t}$) and precision-ordering ($\mathbf{u} \leq s, s \in \{\mathbf{t}, \mathbf{f}\}$) to the complete lattices $\langle Atoms(\mathcal{P})^2, \preceq_t^2\rangle$ and $\langle Atoms(\mathcal{P})^2, \preceq_p^2\rangle$ respectively. We have an element $(T, P) \in \wp(Atoms(\mathcal{P}))^2$ where $T \not\subseteq P$, thus not every element in $\wp(Atoms(\mathcal{P}))^2$ is an interpretation. If we restrict our focus to interpretations (i.e., pairs s.t. $T \subseteq P$), the lattice orderings above compare the individual truth evaluations of atoms. For example, an interpretation that assigns $\mathbf{t}$ to an atom a is greater (for both lattices) than an interpretation that assigns $\mathbf{u}$ to a (assuming other atoms have the same truth values).

We formally define rule satisfaction. Given a pair of sets $A, B \subseteq Atoms(\mathcal{P})$ (not necessarily an interpretation) we define *satisfied*$_{(A,B)}(\mathcal{P})$ as the subset of $\mathcal{P}$, $\{r \in \mathcal{P} \mid$ *body*$^+(r) \subseteq A$, *body*$^-(r) \cap B = \emptyset\}$. For an interpretation (T, P), the set *satisfied*$_{(T,P)}(\mathcal{P})$ contains all rules whose body evaluates as true (satisfied without undefined atoms), while *satisfied*$_{(P,T)}(\mathcal{P})$ contains all rules whose body evaluates as possibly true (satisfied with or without undefined atoms). We say an interpretation (T, P) is a *model* of a program $\mathcal{P}$ if all rules are satisfied by (T, P), i.e., for each $r \in \mathcal{P}$,

$$(r \in \textit{satisfied}_{(T,P)}(\mathcal{P})) \Rightarrow (\textit{head}(r) \cap T \neq \emptyset),\ \text{and}$$
$$(r \in \textit{satisfied}_{(P,T)}(\mathcal{P})) \Rightarrow (\textit{head}(r) \cap P \neq \emptyset)$$

The model condition ensures every satisfied rule has an atom in its head with an appropriate truth value. A stable model minimizes truth while keeping negation stable [19]. Intuitively, we do not want to lower the truth values of atoms if it results in rules deriving higher truth values for some of those atoms. To accommodate this, when shrinking an interpretation (T, P) w.r.t. $\preceq_t^2$, we evaluate the negative bodies of rules against the original interpretation (T, P) while the positive bodies and rule heads are evaluated against the shrunken interpretation (X, Y). We outline this formally in the following. Given two interpretations (X, Y) and (T, P), we say (X, Y) is a model of the reduct of $\mathcal{P}$ w.r.t. (T, P) if for each $r \in \mathcal{P}$,

$$(r \in \textit{satisfied}_{(X,P)}(\mathcal{P})) \Rightarrow (\textit{head}(r) \cap X \neq \emptyset),\ \text{and}$$

$$(r \in \textit{satisfied}_{(Y,T)}(\mathcal{P})) \Rightarrow (\textit{head}(r) \cap Y \neq \emptyset)$$

A model is *stable* if it's the smallest model (w.r.t. truth) of its reduct.

Definition 4.1. *A model (T, P) of a DLP $\mathcal{P}$ is a* (partial) stable model *of $\mathcal{P}$ if there is no model (X, Y) of the reduct of $\mathcal{P}$ w.r.t. (T, P) s.t. $(X, Y) \prec_t^2 (T, P)$.*

Partial stable models correspond to answer sets when $T = P$.

4.2. Preliminaries: Hybrid MKNF Knowledge Bases

Defined by Motik and Rosati [10], hybrid MKNF knowledge bases (HMKNF) are a logic designed to combine ontologies with answer set programming faithfully. Ontologies operate under the open-world assumption where a fact is only false if it can be proven as such (e.g., through classical reasoning). Conversely, answer set programming operates under the closed-world assumption, where facts are assumed to be false if they cannot be proven.

A critical goal of hybrid MKNF is to construct a hybrid logic that does not simply layer ontologies on top of answer set programs or vice-versa. A logic is *tight* [10] if we can perform interleaved reasoning between ontological concepts and rules. For example, because hybrid MKNF satisfies this property, a consequence can follow from something in the ontology which follows from a rule, which follows from the ontology.

In this paper, we focus very little on ontologies and instead treat them as decidable, polynomial first-order formulas akin to reasoning black boxes that can perform an entailment function. As such, we introduce a stripped-down (yet equivalent) version of hybrid MKNF that deals primarily with answer set programs.

We give a summary of the semantics of hybrid MKNF knowledge bases using the three-valued semantics that were introduced by Knorr et al. [20]. A (disjunctive) hybrid MKNF knowledge base $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ (henceforth, a HMKNF KB) is a disjunctive logic program $\mathcal{P}$ coupled with an ontology $\mathcal{O}$ [10]. Here, an ontology $\mathcal{O}$ is simply an object that can be translated to a first-order formula $\pi(\mathcal{O})$, whose entailment relation can be computed in polynomial-time. For an ontology $\mathcal{O}$ and a set of atoms $S \subseteq Atoms(\mathcal{P})$ the entailment relation $\mathsf{OB}_{\mathcal{O},S} \models \phi$ holds for a propositional formula ϕ if ϕ is entailed by $\pi(\mathcal{O}) \wedge (\bigwedge S)$, this relation always holds if $\pi(\mathcal{O}) \wedge (\bigwedge S)$ is inconsistent (the principle of explosion) and $\mathsf{OB}_{\mathcal{O},S} \not\models a$ holds otherwise.

An interpretation (T, P) is $\mathcal{O}$*-consistent* if $\pi(\mathcal{O}) \wedge (\bigwedge P)$ is consistent (i.e. it has at least one model under first-order logic). An $\mathcal{O}$-consistent (T, P) is $\mathcal{O}$*-saturated* if $\mathsf{OB}_{\mathcal{O},P} \not\models a$ for each $a \in (Atoms(\mathcal{P}) \setminus P)$ and $\mathsf{OB}_{\mathcal{O},T} \not\models a$ for each $a \in (Atoms(\mathcal{P}) \setminus T)$[1]. Intuitively, we cannot derive new consequences from $\mathcal{O}$ w.r.t. an $\mathcal{O}$-saturated interpretation. We call an interpretation (T, P) a *model* of a HMKNF KB $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ if it is a model of $\mathcal{P}$ and it is $\mathcal{O}$-saturated.

Definition 4.2. *We call a model (T, P) of an HMKNF KB $\mathcal{K}$ a $\mathcal{P}$-*MKNF model *of $\mathcal{K}$ if for any $\mathcal{O}$-saturated interpretation (X, Y) s.t. $(X, Y) \prec_t^2 (T, P)$, (X, Y) is not a model of the reduct of $\mathcal{P}$ w.r.t. (T, P).*

This definition very similar to partial stable models (Definition 4.1) with the key difference that (T, P) must be $\mathcal{O}$-saturated and all smaller interpretations tested against the reduct must also be $\mathcal{O}$-saturated.

We give an example knowledge base to instantiate and demonstrate the above definitions.

Example 3. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ such that $\pi(\mathcal{O}) = ((c \vee a) \wedge d)$ and $\mathcal{P}$ contains the rule $(a, b \leftarrow)$. The interpretation $(\{b\}, \{b\})$ is not $\mathcal{O}$-saturated because $d \in (Atoms(\mathcal{P}) \setminus T)$ and $\mathsf{OB}_{\mathcal{O},\{b\}} \models d$ (more generally, $\mathsf{OB}_{\mathcal{O},\emptyset} \models d$). The interpretation $(\{b, d\}, \{b, d\})$ is $\mathcal{O}$-saturated. Note that even though c is false w.r.t. $(\{b, d\}, \{b, d\})$ and $\pi(\mathcal{O}) \wedge \neg c$ implies a, we do not derive a because falsely assigned atoms do not participate in the derivation of positive atoms using the ontology. This is a critical aspect of how hyrbid MKNF differentiates between classically false atoms and atoms that are false due to negation as failure.*

[1]We require (T, P) to be $\mathcal{O}$-consistent to handle the case where $P = Atoms(\mathcal{P})$, in all other cases, it is implied by the principle of explosion.

As a result, this knowledge base has two $\mathcal{P}$-MKNF models, namely $(\{b,d\},\{b,d\})$ and $(\{a,d\},\{a,d\})$. Note that $\mathcal{P}$ has two partial stable models $((\{b\},\{b\})$ and $(\{a\},\{a\}))$ and $\mathcal{O}$ simply requires that d be true.

$\mathcal{P}$-MKNF models and partial stable models are closely related. For HMKNF KBs with empty ontologies (every interpretation is $\mathcal{O}$-saturated), $\mathcal{P}$-MKNF models are equivalent to partial stable models. Here, we present $\mathcal{P}$-MKNF models instead of hybrid MKNF models (as introduced by Knorr et al. [20]) because the full introduction of the logic is not necessary for our results. We can limit our focus to the interpretation of atoms that appear in $\mathcal{P}$ because it is straightforward to interpret the remaining atoms from $\mathcal{O}$.

Proposition 4.1. *Originally, hybrid MKNF uses MKNF interpretations, sets of first-order interpretations (interpretations of first-order formulas), to interpret atoms. We can convert a $\mathcal{P}$-MKNF model to a full MKNF model of $\mathcal{K}$ named (M,N).*

$$(M,N) = (\{I \mid \mathsf{OB}_{\mathcal{O},T} \models I\}, \{I \mid \mathsf{OB}_{\mathcal{O},P} \models I\})$$

We can also construct an interpretation (T,P) from a MKNF model (M,N)

$$T \coloneqq \{a \in Atoms(\mathcal{P}) \mid \forall I \in M, I \models a\}$$
$$P \coloneqq \{a \in Atoms(\mathcal{P}) \mid \forall I \in N, I \models a\}$$

A proof of this proposition can be found in [21], which is based on a prior work [12], where we showed how the transformation outlined above is related to the usual formulation of partial MKNF models of hybrid MKNF [20].

In prior work [22], we shown that, using the transformation outlined above, one can easily generalize the work in this paper to the usual formulation of hybrid MKNF [10].

4.3. Applying Approximator Sets

Liu and You [1] introduce an approximator for normal HMKNF KBs. We use *head*$(\mathcal{P})$ to denote the set of all head atoms in the program $\mathcal{P}$, i.e., the set $(\bigcup_{r\in\mathcal{P}} \textit{head}(r))$.

Definition 4.3 (Liu and You [1]). *Given an HMKNF KB $\mathcal{K} = (\mathcal{O},\mathcal{P})$*

$$\Gamma_{\mathcal{P}}(T,P) \coloneqq \textit{head}(\textit{satisfied}_{(T,P)}(\mathcal{P}))$$
$$\Phi_{\mathcal{K}}(T,P)_1 \coloneqq \Gamma_{\mathcal{P}}(T,P) \cup \{a \mid \mathsf{OB}_{\mathcal{O},T} \models a\}$$
$$\Phi_{\mathcal{K}}(T,P)_2 \coloneqq \Big(\Gamma_{\mathcal{P}}(P,T) \setminus \{a \mid \mathsf{OB}_{\mathcal{O},T} \models \neg a\}\Big) \cup \{a \mid \mathsf{OB}_{\mathcal{O},P} \models a\}$$

The above operator is nearly symmetric (i.e. $\Phi_{\mathcal{K}}(T,P)_1 = \Phi_{\mathcal{K}}(P,T)_2$), however, the ontology may be used to block the derivation of some atoms in the heads of rules (Thus, $\Phi_{\mathcal{K}}(T,P)_1 \neq \Phi_{\mathcal{K}}(P,T)_2$). This can result in some stable fixpoints having the property where some rules are not satisfied because the derivation of a head atom is blocked. This can easily be detected by checking whether $\Gamma_{\mathcal{P}}(P,T)_1 \cap \{a \mid \mathsf{OB}_{\mathcal{O},T} \models \neg a\}$ is empty. When this set is empty, $\Phi(T,P)_2$ is equal to $\Phi(P,T)_1$, i.e., the operator is symmetric for the given pair.

We lift this approximator to disjunctive HMKNF KBs by defining an a-set paired with the aforementioned symmetry condition from Liu and You [1].

Definition 4.4. *Let $\mathcal{K} = (\mathcal{O},\mathcal{P})$ be a HMKNF KB. We define $\langle H_{\mathcal{K}}, \Theta_{\mathcal{K}}\rangle$,*

$$H_{\mathcal{K}} \coloneqq \{\Phi_{(\mathcal{O},\mathcal{P}')} \mid \mathcal{P}' \in \textit{normal}(\mathcal{P})\}$$
$$\Theta_{\mathcal{K}} \coloneqq \{(h,(T,P)) \in (H_{\mathcal{K}} \times \mathcal{L}^2) \mid$$
$$h(T,P) = (h(T,P)_1, h(P,T)_1) \text{ and}$$

$$\mathsf{OB}_{\mathcal{O},P} \text{ is consistent } \}$$

$$normal(\mathcal{P}) \coloneqq \Big\{ \{hc(r) \mid r \in \mathcal{P}\} \mid \exists hc, \forall r \in \mathcal{P}, \exists a \in \textit{head}(r),\ hc(r) = (a \leftarrow \textit{body}(r)) \Big\}$$

The set *normal*($\mathcal{P}$) splits a program into its normal counterparts by mapping each disjunctive rule to a normal rule with an overlapping head. This set is a subset of split disjunctive databases [23], which map each rule to several normal rules. This a-set captures three-valued hybrid MKNF semantics.

Theorem 4.1. *A pair* (T, P) *is an a-stable fixpoint of* $\langle H_\mathcal{K}, \Theta_\mathcal{K}\rangle$ *iff it is a* $\mathcal{P}$*-MKNF model of* $\mathcal{K}$.

Proof. ($\Rightarrow$ by contrapositive) Let (T, P) be an interpretation s.t. its not an $\mathcal{P}$-MKNF model of $\mathcal{K}$. If (T, P) is $\mathsf{OB}_{\mathcal{O},P}$ is inconsistent, then $\neg\exists(h, (T, P)) \in S(\Theta_\mathcal{K})$. We assume $\mathsf{OB}_{\mathcal{O},P}$ is consistent as $\neg\exists(h, (T, P)) \in \Theta_\mathcal{K}$ otherwise. If (T, P) is not $\mathcal{O}$-saturated, then either

$$\{a \notin T \mid \mathsf{OB}_{\mathcal{O},T} \models a\} \neq \emptyset, \text{ or}$$
$$\{a \notin P \mid \mathsf{OB}_{\mathcal{O},P} \models a\} \neq \emptyset$$

Thus, (T, P) is not a fixpoint (or stable fixpoint) of any $\Phi \in H_\mathcal{K}$. We assume (T, P) is $\mathcal{O}$-saturated. Suppose (T, P) is not a model of $\mathcal{P}$. It follows that $\mathcal{P}$ is not a model of any $\mathcal{P}' \in normal(\mathcal{P})$. We show $\Phi_{(\mathcal{O},\mathcal{P}')} \notin S(\Theta_\mathcal{K})$

(Case $r \in \textit{satisfied}_{(P,T)}(\mathcal{P})$ s.t. $\{a\} = \textit{head}(r)$ and $\mathsf{OB}_{\mathcal{O},T} \models \neg a$) We have $a \notin \Phi_{(\mathcal{O},\mathcal{P}')}(T, P)_2$ yet $a \in \Phi_{(\mathcal{O},\mathcal{P}')}(P, T)_1$. It follows that $(\Phi_{(\mathcal{O},\mathcal{P}')}, (T, P)) \notin \Theta_\mathcal{K}$. (Case otherwise) We have some rule $r \in \textit{satisfied}_{(P,T)}(\mathcal{P})$ (resp. $r \in \textit{satisfied}_{(T,P)}(\mathcal{P})$) s.t. $\textit{head}(r) \cap P = \emptyset$ (resp. $\textit{head}(r) \cap T = \emptyset$). In either case, (T, P) is not a fixpoint (or stable fixpoint) of $\Phi_{(\mathcal{O},\mathcal{P}')}$. It follows that $(\Phi_{(\mathcal{O},\mathcal{P}')}, (T, P)) \notin S(\Theta_\mathcal{K})$. We can now assume (T, P) is a model of $\mathcal{K}$. Because (T, P) is not a $\mathcal{P}$-MKNF model model of $\mathcal{K}$ while it is $\mathcal{O}$-saturated and a model of (T, P), (T, P) is not a stable model of $\mathcal{P}$, i.e., there exists (T', P') s.t. $(T', P') \preceq_t^2 (T, P)$ and (T', P') is a model of the reduct of $\mathcal{P}$ w.r.t. (T', P'). There exists some $\mathcal{P}' \in normal(\mathcal{P})$ s.t. (T', P') is a model of the reduct of $\mathcal{P}'$ w.r.t. (T, P). (T, P) is a $\prec_t^2$-prefixpoint of $S(\Phi_{(\emptyset,\mathcal{P}')})$ (and also of $S(\Phi_{(\mathcal{O},\mathcal{P}')})$). It follows that (T, P) is not in $\min_{\preceq_t^2}\{S(\phi)(T, P) \mid \phi \in H_\mathcal{K}\}$ and thus it is not an a-stable fixpoint of $H_\mathcal{K}$.

($\Leftarrow$) (By contrapositive) Let (T, P) be an interpretation s.t. it is not an a-stable fixpoint of $H_\mathcal{K}$. If (T, P) is not a model of $\mathcal{P}$ or it is not $\mathcal{O}$-saturated, then it is not a $\mathcal{P}$-MKNF model of $\mathcal{K}$. If we assume these, it follows that (T, P) is $\mathcal{O}$-consistent and thus $(T, P) \in \Theta_\mathcal{K}$. We have some $\mathcal{P}' \in normal(\mathcal{P})$ s.t. $S(\Phi_{(\mathcal{O},\mathcal{P}')})(T, P) \prec_t^2 (T, P)$. It follows that there exists $(T', P') \prec_t^2 (T, P)$ such that (T', P') is $\mathcal{O}$-saturated and a model of the reduct of $\mathcal{P}'$ w.r.t. (T, P). Thus, (T, P) is not a $\mathcal{P}$-MKNF model of $\mathcal{K}$. □

A key factor in the rise of expressivity of DLPs [16] is the possibility for head-cycles (cyclic derivation between atoms that occur in the same rule head) to allow for multiple atoms in the head of a rule to be true. We give an example demonstrating the usage of our constructs using a knowledge base with head cycles.

Example 4. *Let* $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ *s.t.* $\mathcal{O} = (\neg a \vee b') \wedge (a' \vee \neg b)$ *and* $\mathcal{P}$ *is defined as follows.*

$$a, b \leftarrow d \qquad a \leftarrow a'$$
$$d \leftarrow \textbf{not}\, c \qquad b \leftarrow b'$$

Here, both a *and* b *appear together in the head of a rule. If* a *is true, then* b' *follows, then* b*. Similarly, if* b *is true, then* a' *follows, then* a*. Thus, a head cycle exists between* a *and* b*. We have* $H_\mathcal{K} = \{\Phi_{(\mathcal{O},\mathcal{P}_a)}, \Phi_{(\mathcal{O},\mathcal{P}_b)}\}$ *where* $(a \leftarrow d) \in \mathcal{P}_a$ *and* $(b \leftarrow d) \in \mathcal{P}_b$*. We sequentially apply the stable operator* $S(\Phi_{(\mathcal{O},\mathcal{P}_a)})$ *to the* $\preceq_p^2$*-least interpretation* $(\emptyset, \{a, a', b, b', c, d\})$.

$$S(\Phi_{(\mathcal{O},\mathcal{P}_a)})(\emptyset, \{a, a', b, b', c, d\}) = (\emptyset, \{a, a', b, b', d\})$$

(c is removed because it cannot be derived)

$$S(\Phi_{(\mathcal{O},\mathcal{P}_a)})(\emptyset, \{a, a', b, b', d\}) = (\{d, a, b', b, a'\}, \{a, a', b, b', d\})$$

(d and a are derived because **not** *c is true and $\mathcal{P}_a$ "picks" a respectively)*

(b' is derived becaue $\mathsf{OB}_{\mathcal{O},\{a\}} \models b'$ and b is derived from the rule $(b \leftarrow b')$)

(Lastly, a' is derived becaue $\mathsf{OB}_{\mathcal{O},\{b\}} \models a'$)

The interpretation $(\{a, a', b, b', d\}, \{a, a', b, b', d\})$ is a stable fixpoint of $S(\Phi_{(\mathcal{O},\mathcal{P}_a)})$. Using similar steps above, we can show that it is also a stable fixpoint of $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})$. In essence, both $\mathcal{P}_a$ and $\mathcal{P}_b$ agree that both a and b should be both true. That is, $(\{a, a', b, b', d\}, \{a, a', b, b', d\})$ is an a-stable fixpoint of $H_\mathcal{K}$, and by Theorem 4.1, a $\mathcal{P}$-MKNF model of $\mathcal{K}$.

If we change $\mathcal{O}$ so that this head cycle is broken, we can observe the minimality check of an a-stable fixpoint in action. Let $\mathcal{O} = (\neg a \vee b')$. The stable fixpoint of $S(\Phi_{(\mathcal{O},\mathcal{P}_a)})$ remains the same, but the stable fixpoint of $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})$ has a and a' as false. Thus,

$$\mathbf{lfp}_{\preceq_p^2}\left(S(\Phi_{(\mathcal{O},\mathcal{P}_b)})\right) \preceq_t^2 \mathbf{lfp}_{\preceq_p^2}\left(S(\Phi_{(\mathcal{O},\mathcal{P}_a)})\right)$$

and it follows that $\mathbf{lfp}_{\preceq_p^2}\left(S(\Phi_{(\mathcal{O},\mathcal{P}_a)})\right)$ is not an a-stable fixpoint of $H_\mathcal{K}$.

Note that in the above example, if we fixed the acceptance relation to be maximal, i.e. $\Theta_\mathcal{K} = H_\mathcal{K} \times Atoms(\mathcal{P})^2$, then the example remains the same. We demonstrate an interesting edge case of our definitions where the acceptance relation plays a critical role.

Example 5. *Let $\mathcal{K} = (\mathcal{O}, \mathcal{P})$ be the HMKNF KB such that $\pi(\mathcal{O}) := \neg b$ and $\mathcal{P}$ is defined as follows*

$$a, b \leftarrow \mathbf{not}\, c, \mathbf{not}\, d \qquad\qquad c \leftarrow \mathbf{not}\, d \qquad\qquad d \leftarrow \mathbf{not}\, c$$

We have normal($\mathcal{P}$) $= \{\mathcal{P}_a, \mathcal{P}_b\}$ where $(a \leftarrow \mathbf{not}\, c, \mathbf{not}\, d) \in \mathcal{P}_a$ and $(b \leftarrow \mathbf{not}\, c, \mathbf{not}\, d) \in \mathcal{P}_b$. This knowledge base has one $\mathcal{P}$-MKNF model which is $(\emptyset, \{a, c, d\})$, that is, the interpretation that assigns a, c, and d to be undefined. This interpretation is $\mathcal{O}$-saturated ($\pi(\mathcal{O}) \wedge \{a, c, d\}$ is consistent, and no positive conclusions can be drawn from the ontology), and all the rules are satisfied.

Because $\mathcal{P}$ induces two normal programs in normal($\mathcal{P}$), the set $H_{\Phi_\mathcal{K}}$ contains two approximators.

$$\{\Phi_{(\mathcal{O},\mathcal{P}_a)}, \Phi_{(\mathcal{O},\mathcal{P}_b)}\} := H_{\Phi_\mathcal{K}}$$

Notably, we have $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})(\emptyset, \{a, c, d\}) = (\emptyset, \{c, d\})$. Without the acceptance relation, this prefixpoint of $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})$ would prevent $(\emptyset, \{a, c, d\})$ from being an a-stable fixpoint (due to $\{c, d\} \subseteq \{a, c, d\}$).

However, we have $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})(\{a, c, d\}, \emptyset)_1 = \{b, c, d\}$ which is not equal to $S(\Phi_{(\mathcal{O},\mathcal{P}_b)})(\emptyset, \{a, c, d\})_2$, thus $(\Phi_{(\mathcal{O},\mathcal{P}_b)}, (\emptyset, \{b, c, d\})) \notin \Theta_\mathcal{K}$ and $\mathcal{P}_b$ does not participate in determining whether $(\emptyset, \{a, c, d\})$ is an a-stable fixpoint.

We have shown that the extension of AFT proposed by Bi et al. [11] can be lifted to disjunctive HMKNF KBs with relative ease using a-sets.

5. Discussion and Future Work

Incorporating classical reasoning into answer set programming creates numerous complications for the use of AFT. We navigate these challenges by adopting a generalized variant of AFT that works with inconsistent pairs [11], and by working with families of approximators to capture nondeterminism. We arrive at a general-purpose framework that can be applied to characterize nonmonotonic, nondeterministic systems that incorporate external reasoning, such as hybrid MKNF. Our framework provides a method to reveal new insights into well-founded propagation which could benefit solvers and grounders for hybrid MKNF. We believe the framework presented in this paper could be applied to other hybrid reasoning semantics, but we leave this for future work.

This work builds upon our previous developments [12] in which we leveraged families of fixpoint operators to capture the three-valued partial stable semantics of hybrid MKNF knowledge bases. Now that we have fully linked the idea of using families of operators to AFT, further advancements with normal hybrid MKNF knowledge bases can lifted to support disjunctive rules. This includes stronger approximators constructed for normal hybrid MKNF knowledge bases such as our recent approximator [22] (which wasn't presented here due to complexity). Our approach to capture disjunctive hybrid MKNF bears some resemblence to the split programs of Sakama [23]. Our mapping to normal rules is more strict and we capture partial stable models [17]. While a possible approach could have been to adopt nondeterministic AFT, which defines nondeterministic stable revision, our approach here has the advantage that we can leverage existing definitions of traditional AFT, whereas nondeterministic AFT is a standalone theory. Further, it's unclear how to lift generalized approximators to nondeterministic AFT, as nondeterministic AFT does not handle inconsistent pairs.

To summarize, we have presented a framework that lifts generalized, inconsistent-capable AFT to handle nondeterministic semantics and we've demonstrated that this framework is useful by applying it to disjunctive hybrid MKNF knowledge bases.

6. Acknowledgments

We would like to thank and acknowledge Alberta Innovates and Alberta Advanced Education for their financial support of author one.

References

[1] F. Liu, J.-H. You, Alternating fixpoint operator for hybrid MKNF knowledge bases as an approximator of AFT, Theory Pract. Log. Program. 22 (2022) 305–334. URL: https://doi.org/10.1017/S1471068421000168. doi:10.1017/S1471068421000168.

[2] J. Heyninck, O. Arieli, B. Bogaerts, Non-deterministic approximation fixpoint theory and its application in disjunctive logic programming, Artificial Intelligence 331 (2024) 104110. URL: https://www.sciencedirect.com/science/article/pii/S0004370224000468. doi:https://doi.org/10.1016/j.artint.2024.104110.

[3] M. Denecker, V. Marek, M. Truszczyński, Approximations, stable operators, well-founded fixpoints and applications in nonmonotonic reasoning, in: Logic-Based Artificial Intelligence, Springer, 2000, pp. 127–144. doi:10.1007/978-1-4615-1567-8_6.

[4] N. Pelov, M. Denecker, M. Bruynooghe, Well-founded and stable semantics of logic programs with aggregates, Theory and Practice of Logic Programming 7 (2007) 301–353.

[5] B. Bogaerts, J. Jansen, B. D. Cat, G. Janssens, M. Bruynooghe, M. Denecker, Bootstrapping inference in the IDP knowledge base system, New Generation Computing 34 (2016) 193–220.

[6] C. Antic, T. Eiter, M. Fink, Hex semantics via approximation fixpoint theory, in: Proc. 12th International Conference on Logic Programming and Nonmonotonic Reasoning, LPNMR-13, Corunna, Spain, 2013, pp. 102–115.

[7] J. Vennekens, D. Gilis, M. Denecker, Splitting an operator: Algebraic modularity results for logics with fixpoint semantics, ACM Transactions on Computational Logic 7 (2006) 765–797.

[8] B. Bogaerts, L. Cruz-Filipe, Fixpoint semantics for active integrity constraints, Artificial Intellegence 255 (2018) 43–70.

[9] B. Bogaerts, J. Vennekens, M. Denecker, Safe inductions and their applications in knowledge representation, Artificial Intelligence 259 (2018) 167–185.

[10] B. Motik, R. Rosati, Reconciling description logics and rules, J. ACM 57 (2010) 30:1–30:62. doi:10.1145/1754399.1754403.

[11] Y. Bi, J.-H. You, Z. Feng, A generalization of approximation fixpoint theory and application, in: R. Kontchakov, M.-L. Mugnier (Eds.), 8th International Conference on Web Reasoning and Rule Systems, RR 2014, Lecture Notes in Computer Science 8741, Springer, Cham, 2014, pp. 45–59.

[12] S. Killen, J.-H. You, A fixpoint characterization of three-valued disjunctive hybrid MKNF knowledge bases, in: Y. Lierler, J. F. Morales, C. Dodaro, V. Dahl, M. Gebser, T. Tekle (Eds.), 38th International Conference on Logic Programming, Technical Communications / Doctoral Consortium, Haifa, Israel, August 2022, volume 364 of *EPTCS*, 2022, pp. 51–64. URL: https://doi.org/10.4204/EPTCS.364.6. doi:10.4204/EPTCS.364.6.

[13] S. Roman, Lattices and Ordered Sets, Springer New York, 2008. URL: http://link.springer.com/10.1007/978-0-387-78901-9. doi:10.1007/978-0-387-78901-9.

[14] A. Tarski, A lattice-theoretical fixpoint theorem and its applications., Pacific Journal of Mathematics 5 (1955) 285 – 309. doi:10.2140/pjm.1955.5.285.

[15] N. Leone, P. Rullo, F. Scarcello, Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation 135 (????) 69–112. URL: https://www.sciencedirect.com/science/article/pii/S0890540197926304. doi:https://doi.org/10.1006/inco.1997.2630.

[16] T. Eiter, G. Gottlob, On the computational cost of disjunctive logic programming: Propositional case, Ann. Math. Artif. Intell. 15 (1995) 289–323. URL: https://doi.org/10.1007/BF01536399. doi:10.1007/BF01536399.

[17] T. C. Przymusinski, Stable semantics for disjunctive programs, New Gener. Comput. 9 (1991) 401–424. doi:10.1007/BF03037171.

[18] N. D. Belnap, A Useful Four-Valued Logic, Springer Netherlands, Dordrecht, 1977, pp. 5–37. URL: https://doi.org/10.1007/978-94-010-1161-7_2. doi:10.1007/978-94-010-1161-7_2.

[19] D. Pearce, From Here to There: Stable Negation in Logic Programming, Springer Netherlands, Dordrecht, 1999, pp. 161–181. URL: https://doi.org/10.1007/978-94-015-9309-0_8. doi:10.1007/978-94-015-9309-0_8.

[20] M. Knorr, J. J. Alferes, P. Hitzler, Local closed world reasoning with description logics under the well-founded semantics, Artif. Intell. 175 (2011) 1528–1554. doi:10.1016/j.artint.2011.01.007.

[21] R. Kinahan, S. Killen, K. Wan, J.-H. You, On the foundations of conflict-driven solving for hybrid MKNF knowledge bases, in: International Conference on of Logic Programming (ICLP 2024), Cambridge, 2024. To appear in a special issue of TPLP.

[22] S. Killen, W. Gao, J.-H. You, Expanding the class of polynomial time computable well-founded semantics for hybrid MKNF, in: Workshops co-located with the 39th International Conference on Logic Programming (ICLP 2023), London, United Kingdom, July 2023, volume 3437 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3437/paper8ASPOCP.pdf.

[23] C. Sakama, Possible model semantics for disjunctive databases, in: Deductive and Object-Oriented Databases, Elsevier, 1990, pp. 369–383.