What If No Hybrid Reasoner Is Available?

Hybrid MKNF in Multi-Context Systems

Matthias Knorr[1], Martin Slota[1], Martin Homola[2], and João Leite[1]

[1]CENTRIA & Departamento de Informática, Universidade Nova de Lisboa,
Quinta da Torre, 2829-516 Caparica, Portugal
[2]Faculty of Mathematics, Physics and Informatics, Comenius University,
Mlynská dolina, 842 48 Bratislava, Slovakia

Abstract

In open environments, agents need to reason with knowledge from various sources, represented in different languages. Multi-Context Systems (MCSs) allow for the integration of knowledge from different heterogeneous sources in an effective and modular way. Whereas most knowledge bases (contexts) typically considered within an MCS are written in some description logic or some non-monotonic rule-based language, sometimes more expressive languages that combine the features of both these paradigms are necessary, such as Hybrid MKNF. However, since agents may not have access to specialised reasoners for contexts using all these languages, it proves useful to have tools that equivalently simplify or transform a given MCS into another MCS that only uses the reasoners that are available.

In this paper, we thoroughly investigate the relation between MCSs and Hybrid MKNF. We provide a number of transformations that show that Hybrid MKNF knowledge bases can be embedded into MCSs without the need for specific MKNF reasoners. To complete the picture, we also show that when an MKNF reasoner is available, it can be used to handle several description logic and rule contexts joined into a single MKNF context. Furthermore, we show that we can encapsulate the non-monotonic transfer of information between different contexts in one rule language context, allowing e.g. the use of external non-monotonic reasoners.

1 Introduction

In *Open Multi-Agent Systems*, the paradigm for knowledge representation and reasoning is rapidly changing from one where each agent has its own monolithic knowledge base written in some language, for which the agent has a specific reasoner, into one where agents have to deal with several external heterogeneous knowledge sources, possibly written in different languages. These sources of knowledge include the increasing number of available ontologies and rule sets, to a large extent developed within initiatives such as Semantic Web and Linked Open Data, the norms and policies to promote desirable general properties published by the institutions that increasingly govern agent interaction and cooperation,[1] and the information communicated by other agents. Making sense of the knowledge obtained from all these different sources is crucial to

[1] See, e.g., http://www.normativemas.org/ and http://www.deonticlogic.org/.

increase the chance of individually making the right choice and potentiate the chance of agreement in negotiations. But, to make sense of all these different sources of knowledge, agents need different specialised reasoners to properly understand their meaning, and also ways to deal with the possible interactions between knowledge with different provenance.

To deal with such diverse sources of knowledge, agent developers have turned their attention to *Multi-Context Systems* (MCS) [4, 5, 8, 9, 21, 23]. Within MCSs, knowledge is modularly composed of contexts, each of which possibly encapsulating a source of knowledge of a different type, while bridge rules provide effective means for integration [13, 14]. With the equilibria semantics of Brewka and Eiter [6], MCSs provide an effective and modular way to integrate knowledge from different heterogeneous sources, for example, different ontologies written in some Description Logic (DL) based ontology language, such as OWL, a rule set written in Answer-Set Programming (ASP) representing some business policies, or some facts written in propositional logic representing the agent's model of some other agent, to name only a few. MCSs are simple enough to allow this heterogeneous knowledge to be bridged and integrated, while keeping their distinct provenance.

For example, consider an airport and some agent in charge of its security. First of all, there are many *ontologies* available that describe airports (e.g. that airports have terminals, that terminals have gates, etc.) which could be (re-)used by our security agent, avoiding the need to develop and maintain them. Then, there are the *National Airport Security Norms*, published by some national authority, that describe what is allowed, forbidden, expected, etc., with respect to airport security (e.g. that a person without a valid passport is not allowed inside the international terminal, with the exception of authorised airport personnel, etc.). In order to enforce security in general, and these norms in particular, each individual airport has a set of *policies* that prescribe how to deal with the situations that may occur (e.g. that passengers on some black list should be interviewed prior to being cleared in the security check). In order to identify occurring situations, our security agent relies on the information incoming from existing *sensors*, which can be more or less complex (e.g. with facial identification, population density, etc.). Finally, as in any true multi-agent system, we can assume the existence of other agents, each of which our agent will have a *model* of.[2]

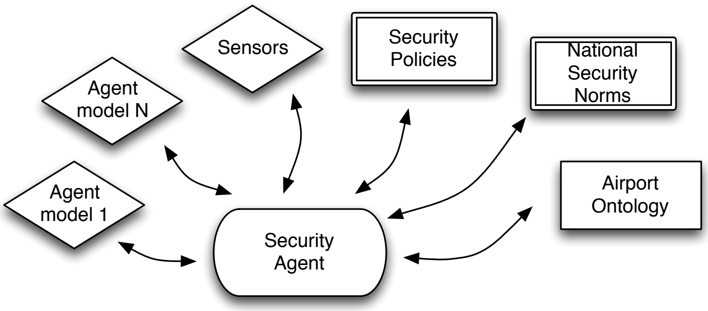

Figure 1: MCS representing the agent's view of the system

[2] In a more realistic distributed setting, beyond the point we try to make in this paper, there would be more than one Security Agent, each with its own MCS, using the same Ontology, Norms and Policies, and perhaps using different contexts for the sensors and models of other agents.

In such an MCS, ontologies can be modelled with DL contexts, the policies and norms with rule-based contexts, for example using logic programs, other agents with separate contexts using propositional facts or rules in a more complex case, and sensors with propositional facts or DL ABox assertions. To propagate logical consequences across contexts, contexts are connected with bridge rules, illustrated by arrows in Fig. 1.

Recently, it has been shown in [1, 2] that realistic norms and policies that mimic the real world require a more complex knowledge representation formalism, such as Hybrid MKNF [19] – based on the Logic of Minimal Knowledge and Negation as Failure (MKNF) [18] – that tightly combines Logic Programming (LP) and Description Logic. In such scenarios, the Closed World Assumption provided by LP rules is used, for example, to deal with defeasible knowledge, such as exceptions, while the Open World Assumption provided by DL axioms is employed e.g. to deal with ontological knowledge and features such as reasoning with unknown individuals. In the context of our airport example, a policy specifying e.g. that *whenever two known suspects (belonging to some black list) are together with a group of at least 5 (possibly unknown) people, then a security alert should be issued, unless the two suspects are diplomats*, would require a hybrid language that allows the joint use of defaults, of LP rules, to deal with the exception for the case when the suspects are diplomats, and existentials, of DL, to deal with unknown individuals (recall that LP cannot reason about unknown individuals, see e.g. Sect. 4.3 in [22]).

If norms and policies are to be published in Hybrid MKNF, two main questions arise, namely:

- Can we use these Hybrid MKNF knowledge bases as contexts in an MCS?
- Will agents require a specialised reasoner to deal with them?

Whereas it would be quite important to be able to use knowledge bases written in these very expressive languages within MCSs, it is quite expectable that many agents would not have access to a specialised reasoner to deal with them. Furthermore, whereas there are many efficient reasoners for various kinds of DLs [20] and logic programming languages such as ASP [11], the state of the art w.r.t. reasoners for Hybrid MKNF is still limited to [15] and [16].

To answer these two main questions, in this technical paper we investigate the relationship between MCSs and Hybrid MKNF. Taking the two-valued semantics of Hybrid MKNF [19], which is based on the Answer Set Semantics [12], we formally show that

- Hybrid MKNF knowledge bases can be used in the form of an MKNF context, which can then be bridged to other contexts, so we can use the expressivity of Hybrid MKNF within MCSs.
- Hybrid MKNF knowledge bases can be embedded into MCSs without the need for a specific MKNF reasoner, in one of the following linearly implementable ways:
  - by translating them each into a first-order context, with non-monotonic bridge rules, so that the agent would need a first-order reasoner;
  - by translating them each into a DL context and a relational database (or fact base) context, with non-monotonic bridge rules, so the agent would *only* need a DL solver and a database lookup process while the rules would be handled by the MCS;
  - by translating them each into a DL context and an ASP context, with mono-

tonic bridge rules, so the agent would need a DL solver and an ASP solver for the non-monotonic ASP context, which could be outsourced to an external service because the bridge rules are monotonic.

We also establish the following results between MCSs and Hybrid MKNF:

- several DL, Database, ASP, and MKNF contexts in an MCS can be joined into one MKNF context/KB, which can be useful in case the agent only has a hybrid MKNF reasoner and, more importantly, is important from a conceptual/theoretical point of view, because it presents the converse of the previous results;
- all non-monotonic bridge rules in an arbitrary MCS can be transferred into one ASP context, leaving only monotonic bridge rules, so a source of computational complexity is removed and the non-monotonic computation can be outsourced to some external service.

The remainder of this paper is structured as follows. After recalling required material in Sect. 2, in Sect. 3 we formally define the contexts we are going to use, followed by an example scenario in Sect. 4. Then, we present the transformations from MKNF contexts into other contexts in Sect. 5, and the converse transformation in Sect. 6. We conclude in Sect. 7 and point to future directions. All the proofs are contained in a separate appendix.

2 Preliminaries

2.1 Description Logics

We first briefly summarise the syntax and semantics of function-free first-order logic with equality which forms the basis for representing both ontological and rule-based knowledge. We assume the standard syntax of first-order *atoms*, *formulas*, and *sentences*, defined inductively over disjoint sets of *constant* and *predicate symbols* $\mathbf{C}$ and $\mathbf{P}$. A first-order formula is *ground* if it contains no variables. The set of all first-order sentences is denoted by Φ. A *first-order theory* is a set of first-order sentences. In the semantics we consider first-order interpretations under the unique name assumption (UNA). The satisfaction of a first-order sentence ϕ in such an interpretation I is denoted by $I \models \phi$; we also say that I is a *model of* ϕ if $I \models \phi$.

Description Logics (DLs) [3] are fragments of first-order logic for which the standard reasoning tasks, such as satisfiability, are usually decidable. We assume that some first-order fragment is used to describe an *ontology*, i.e. to specify a shared conceptualisation of a domain of interest. Unless stated otherwise, we do not constrain ourselves to a specific DL for representing ontologies. The only assumption made in the theoretical developments is that the ontology language is a syntactic variant of a fragment of first-order logic. We assume that for any ontology axiom ϕ and ontology $\mathcal{O}$, $\kappa(\phi)$ and $\kappa(\mathcal{O})$ denote a first-order sentence that semantically corresponds to ϕ and $\mathcal{O}$, respectively. Such translations are known for most DLs [3]. Given a first-order sentence ϕ, we say that an ontology $\mathcal{O}$ *entails* ϕ, denoted by $\mathcal{O} \models \phi$, if every first-order model of $\kappa(\mathcal{O})$ is also a first-order model of ϕ.

2.2 Logic Programs

Like Description Logics, Logic Programming has its roots in classical first-order logic. However, logic programs diverge from first-order semantics by adopting the Closed

World Assumption and allowing for non-monotonic inferences. In what follows, we introduce the syntax of extended normal logic programs and define the *answer set semantics* [12] for such programs.

Syntactically, logic programs are built from *atoms* consisting of first-order atoms without equality. An *objective literal* is an atom p or its (strong) negation $\neg p$. We denote the set of all objective literals by $\mathbf{L}$ and the set of ground objective literals by $\mathbf{L}_\mathsf{G}$. A *default literal* is an objective literal preceded by $\sim$ denoting *default negation*. A *literal* is either an objective literal or a default literal. Given a set of literals B, we introduce the following notation: $B^+ = \{ l \in \mathbf{L} \mid l \in B \}$, $B^- = \{ l \in \mathbf{L} \mid {\sim} l \in B \}$, ${\sim} B = \{ {\sim} L \mid L \in B \}$.

A *rule* is a pair $\pi = (H(\pi), B(\pi))$ where $H(\pi)$ is an objective literal, referred to as *head of* π, and $B(\pi)$ is a set of literals, referred to as *body of* π. Usually, for convenience, we write π as $(H(\pi) \leftarrow B(\pi)^+, {\sim} B(\pi)^-.)$ A rule is called *ground* if it does not contain variables and *definite* if it does not contain any default literal. The *grounding* of a rule π is the set of rules $\mathsf{gr}(\pi)$ obtained by replacing in π all variables with constant symbols from $\mathbf{C}$ in all possible ways. A *program* is a set of rules. A program is *ground* if all its rules are ground; *definite* if all its rules are definite. The grounding of a program $\mathcal{P}$ is defined as $\mathsf{gr}(\mathcal{P}) = \bigcup_{\pi \in \mathcal{P}} \mathsf{gr}(\pi)$.

The answer sets of a program are determined by considering its first-order models in which all constant symbols are interpreted by themselves. An interpretation thus corresponds to a subset of $\mathbf{L}_\mathsf{G}$. Following [12], an answer set is either an interpretation that does not contain both p and $\neg p$ for any ground atom p or, if the considered program is inconsistent, $\mathbf{L}_\mathsf{G}$. An answer set is a model of the program that can be fully derived using rules of the program assuming that literals not present in the model are false.

Definition 1. Let $\mathcal{P}$ be a ground program. An interpretation J is an *answer set of* $\mathcal{P}$ if J is the smallest subset of $\mathbf{L}_\mathsf{G}$ that is equal to $\mathbf{L}_\mathsf{G}$ in case it contains both p and $\neg p$ for some ground atom p, and satisfies all rules of the *Gelfond-Lifschitz reduct of* $\mathcal{P}$ *relative to* J, obtained from $\mathcal{P}$ by deleting all

1. rules with ${\sim} l$ in the body such that $J \models l$, and
2. default literals from the bodies of remaining rules.

The answer sets of a non-ground program $\mathcal{P}$ are the answer sets of $\mathsf{gr}(\mathcal{P})$.

2.3 (Hybrid) MKNF Knowledge Bases

(Hybrid) MKNF Knowledge Bases [19] can be used to join DL knowledge bases and logic programs in a seamless way. They are based on the logic of Minimal Knowledge and Negation as Failure (MKNF) [18], an extension of first-order logic with two modal operators: $\mathbf{K}$ and $\mathbf{not}$. *MKNF sentences* and *theories* are defined by extending function-free first-order syntax by the mentioned modal operators in a natural way.

The semantics of MKNF theories is determined by Herbrand interpretations over a set of constants $\mathbf{C}^*$ that includes $\mathbf{C}$ and in addition contains an infinite supply of constants that do not belong to $\mathbf{C}$. Also, equality is interpreted as identity on $\mathbf{C}^*$.[3] The set of all such Herbrand interpretations is denoted by $\mathbf{I}$. An *MKNF structure* is a triple $(I, \mathcal{M}, \mathcal{N})$ where $I \in \mathbf{I}$ and $\mathcal{M}, \mathcal{N} \subseteq \mathbf{I}$. Intuitively, the first component is used to interpret the first-order parts of an MKNF sentence while the other two components

[3] In [19], equality is treated slightly differently by interpreting it as a congruence relation. Here we adopt the UNA instead. This is in line with assumptions common in Description Logics (see [3]) and ASP, and avoids some additional technicalities in the subsequent sections.

interpret the $\mathbf{K}$ and $\mathbf{not}$ modalities, respectively. By $\phi[a/x]$ we denote the formula obtained from ϕ by replacing every unbound occurrence of variable x with the constant symbol a. Satisfaction of an MKNF sentence and an MKNF theory $\mathcal{T}$ in $(I, \mathcal{M}, \mathcal{N})$ is defined as follows:

$$\begin{array}{lll}
(I,\mathcal{M},\mathcal{N}) \models p & \text{iff} & I \models p \\
(I,\mathcal{M},\mathcal{N}) \models \neg\phi & \text{iff} & (I,\mathcal{M},\mathcal{N}) \not\models \phi \\
(I,\mathcal{M},\mathcal{N}) \models \phi_1 \wedge \phi_2 & \text{iff} & (I,\mathcal{M},\mathcal{N}) \models \phi_1 \text{ and } (I,\mathcal{M},\mathcal{N}) \models \phi_2 \\
(I,\mathcal{M},\mathcal{N}) \models \exists x : \phi & \text{iff} & (I,\mathcal{M},\mathcal{N}) \models \phi[a/x] \text{ for some } a \in \mathbf{C}^* \\
(I,\mathcal{M},\mathcal{N}) \models \mathbf{K}\,\phi & \text{iff} & (J,\mathcal{M},\mathcal{N}) \models \phi \text{ for all } J \in \mathcal{M} \\
(I,\mathcal{M},\mathcal{N}) \models \mathbf{not}\,\phi & \text{iff} & (J,\mathcal{M},\mathcal{N}) \not\models \phi \text{ for some } J \in \mathcal{N} \\
(I,\mathcal{M},\mathcal{N}) \models \mathcal{T} & \text{iff} & (I,\mathcal{M},\mathcal{N}) \models \phi \text{ for all } \phi \in \mathcal{T}
\end{array}$$

The symbols $\top$, $\bot$, $\vee$, $\forall$ and $\supset$ are interpreted accordingly. Also, for any $\mathcal{M} \subseteq \mathbf{I}$, we write $\mathcal{M} \models \mathcal{T}$ if $(I, \mathcal{M}, \mathcal{M}) \models \mathcal{T}$ for all $I \in \mathcal{M}$. An *MKNF interpretation* $\mathcal{M}$ is a non-empty subset of $\mathbf{I}$. The semantics of MKNF theories is defined as follows:

Definition 2. Let $\mathcal{T}$ be an MKNF theory. We say that an MKNF interpretation $\mathcal{M}$ is

- an *S5 model of* $\mathcal{T}$ if $\mathcal{M} \models \mathcal{T}$;
- an *MKNF model of* $\mathcal{T}$ if $\mathcal{M}$ is an S5 model of $\mathcal{T}$, and, for every MKNF interpretation $\mathcal{M}' \supsetneq \mathcal{M}$, there is some $I' \in \mathcal{M}'$ such that $(I', \mathcal{M}', \mathcal{M}) \not\models \mathcal{T}$.

MKNF knowledge bases [19] consist of two components – an ontology $\mathcal{O}$ and a program $\mathcal{P}$ – and their semantics is given by translation to an MKNF theory. In the following we introduce the syntax and semantics of MKNF knowledge bases in which we constrain the program component to a non-disjunctive logic program.

An *MKNF knowledge base* is a set $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ where $\mathcal{O}$ is an ontology and $\mathcal{P}$ is a logic program. An MKNF knowledge base is *ground* if $\mathcal{P}$ is ground; *definite* if $\mathcal{P}$ is definite. The grounding of an MKNF knowledge base $\mathcal{K}$ is defined as $\mathsf{gr}(\mathcal{K}) = \mathcal{O} \cup \mathsf{gr}(\mathcal{P})$.

The translation function κ is defined for all objective literals l, default literals $\sim l$, sets of literals B, rules π with vector of free variables $\vec{x}$, programs $\mathcal{P}$, and MKNF knowledge bases $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ as follows: $\kappa(l) = \mathbf{K}\,l$, $\kappa(\sim l) = \mathbf{not}\,l$, $\kappa(B) = \bigwedge \{ \kappa(L) \mid L \in B \}$, $\kappa(\pi) = (\kappa(B(\pi)) \supset \kappa(H(\pi)))$, $\kappa(\mathcal{P}) = \{ \kappa(\pi) \mid \pi \in \mathcal{P} \}$ and $\kappa(\mathcal{K}) = \{ \kappa(\mathcal{O}) \} \cup \kappa(\mathcal{P})$. The semantics of MKNF knowledge bases is defined as follows:

Definition 3. Let $\mathcal{K}$ be an MKNF knowledge base. We say that an MKNF interpretation $\mathcal{M}$ is an *S5 model of* $\mathcal{K}$ if $\mathcal{M}$ is an S5 model of $\kappa(\mathcal{K})$. Similarly, $\mathcal{M}$ is an *MKNF model of* $\mathcal{K}$ if $\mathcal{M}$ is an MKNF model of $\kappa(\mathcal{K})$.

2.4 Multi-Context Systems

Following [6], a multi-context system (MCS) consists of a collection of components, each of which contains knowledge represented in some *logic*, defined as a triple $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ where $\mathbf{KB}$ is the set of well-formed knowledge bases of L, $\mathbf{BS}$ is the set of possible belief sets, and $\mathbf{ACC} : \mathbf{KB} \to 2^{\mathbf{BS}}$ is a function describing the semantics of L by assigning to each knowledge base a set of acceptable belief sets. We assume that each element of $\mathbf{KB}$ and $\mathbf{BS}$ is a set and that $\mathbf{BS}$ forms a complete lattice w.r.t. set inclusion.

In addition to the knowledge base in each component, *bridge rules* are used to interconnect the components, specifying what knowledge to assert in one component given certain beliefs held in the other components. Formally, for a collection of logics $L = \langle L_1, \dots, L_n \rangle$, an L_i-*bridge rule* σ *over* L, $1 \leq i \leq n$, is of the form $(H(\sigma) \leftarrow B(\sigma).)$, where $B(\sigma)$ is a set of *bridge literals* of the forms $(r : p)$ and $\mathbf{not}\,(r : p)$ where $1 \leq r \leq n$ and p is an element of some belief set of L_r, and, for each $kb \in \mathbf{KB}_i$, $kb \cup \{ H(\sigma) \} \in \mathbf{KB}_i$. A bridge rule is called *monotonic* if it does not contain bridge literals of the form $\mathbf{not}\,(r : p)$, and *non-monotonic* otherwise.

Putting these concepts together, a *multi-context system* is a sequence of contexts $M = \langle C_1, \dots, C_n \rangle$ where $C_i = (L_i, kb_i, br_i)$, $L_i = (\mathbf{KB}_i, \mathbf{BS}_i, \mathbf{ACC}_i)$ is a logic, $kb_i \in \mathbf{KB}_i$ a knowledge base, and br_i is a set of L_i-bridge rules over $\langle L_1, \dots, L_n \rangle$.

In the following, we present the *grounded equilibria* semantics for MCSs [6] which is inspired by the answer set semantics for logic programs.

For an MCS $M = \langle C_1, \dots, C_n \rangle$, a *belief state of* M is a sequence $S = \langle S_1, \dots, S_n \rangle$ such that each S_i is an element of $\mathbf{BS}_i$. For a bridge literal $(r : p)$ we write $S \models (r : p)$ if $p \in S_r$ and $S \models \mathbf{not}\,(r : p)$ if $p \notin S_r$; for a set of bridge literals B, $S \models B$ if $S \models L$ for every $L \in B$.

A belief state $S = \langle S_1, \dots, S_n \rangle$ of M is an *equilibrium* if, for all i with $1 \leq i \leq n$, the following condition holds:

$$S_i \in \mathbf{ACC}_i(kb_i \cup \{ H(\sigma) \mid \sigma \in br_i \wedge S \models B(\sigma) \}) \ .$$

We say that an equilibrium S is *minimal* if there is no equilibrium $S' = \langle S'_1, \dots, S'_n \rangle$ such that $S'_i \subseteq S_i$ for all i with $1 \leq i \leq n$ and $S'_j \subsetneq S_j$ for some j with $1 \leq j \leq n$.

Now we formalise the conditions under which the minimal equilibrium is *unique*, in which case we assign it as the *grounded equilibrium* of the MCS. This can be guaranteed if the contexts can be *reduced*, using a reduction function, to monotonic ones. Formally, a logic $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ is *monotonic* if

1. $\mathbf{ACC}(kb)$ is a singleton set for each $kb \in \mathbf{KB}$, and
2. $S \subseteq S'$ whenever $kb \subseteq kb'$, $\mathbf{ACC}(kb) = \{ S \}$, and $\mathbf{ACC}(kb') = \{ S' \}$.

Furthermore, $L = (\mathbf{KB}, \mathbf{BS}, \mathbf{ACC})$ is *reducible* if for some $\mathbf{KB}^* \subseteq \mathbf{KB}$ and some reduction function $red : \mathbf{KB} \times \mathbf{BS} \to \mathbf{KB}^*$,

1. the restriction of L to $\mathbf{KB}^*$ is monotonic,
2. for each $kb \in \mathbf{KB}$, and all $S, S' \in \mathbf{BS}$:
   - $red(kb, S) = kb$ whenever $kb \in \mathbf{KB}^*$,
   - $red(kb, S) \subseteq red(kb, S')$ whenever $S' \subseteq S$,
   - $S \in \mathbf{ACC}(kb)$ iff $\mathbf{ACC}(red(kb, S)) = \{ S \}$.

A context $C = (L, kb, br)$ is *reducible* if its logic L is reducible and, for all $H \subseteq \{ H(\sigma) \mid \sigma \in br \}$ and all belief sets S, $red(kb \cup H, S) = red(kb, S) \cup H$.

An MCS is *reducible* if all of its contexts are. Note that a context is reducible whenever its logic L is monotonic. In this case $\mathbf{KB}^*$ coincides with $\mathbf{KB}$ and *red* is identity with respect to the first argument. A reducible MCS $M = \langle C_1, \dots, C_n \rangle$ is *definite* if

1. all bridge rules in all contexts are monotonic,
2. for all i and all $S \in \mathbf{BS}_i$, $kb_i = red_i(kb_i, S)$.

In a definite MCS, bridge rules are monotonic, and knowledge bases are already in reduced form. Inference is thus monotonic and a unique minimal equilibrium exists. We take this equilibrium to be the grounded equilibrium:

Definition 4. Let M be a definite MCS. A belief state S of M is the *grounded equilibrium of* M, denoted by $\mathbf{GE}(M)$, if S is the unique minimal equilibrium of M.

Grounded equilibria for general MCSs are defined based on a reduct which generalises the Gelfond-Lifschitz reduct to the multi-context case:

Definition 5. Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible MCS and $S = \langle S_1, \ldots, S_n \rangle$ a belief state of M. The *S-reduct of* M is $M^S = \langle C_1^S, \ldots, C_n^S \rangle$ where, for each $C_i = (L_i, kb_i, br_i)$, we define $C_i^S = (L_i, red_i(kb_i, S_i), br_i^S)$. Here, br_i^S results from br_i by deleting all

1. rules with $\mathbf{not}\ (r : p)$ in the body such that $S \models (r : p)$, and
2. $\mathbf{not}$ literals from the bodies of remaining rules.

For each MCS M and each belief set S, the S-reduct of M is definite. We can thus check whether S is a grounded equilibrium in the usual manner:

Definition 6. Let M be a reducible MCS. A belief state S of M is a *grounded equilibrium of* M if S is the grounded equilibrium of M^S, that is $S = \mathbf{GE}(M^S)$.

3 Concrete Contexts

At first, we define an MKNF context and show that it is reducible thus showing that Hybrid MKNF knowledge bases can indeed be used within MCS. We also formally define a number of related contexts we are going to use throughout the paper.

We start with a *first-order context*.

The *first-order logic* is the logic $L_{\text{FO}} = (\mathbf{KB}_{\text{FO}}, \mathbf{BS}_{\text{FO}}, \mathbf{ACC}_{\text{FO}})$ where

- $\mathbf{KB}_{\text{FO}}$ is the set of first-order theories,
- $\mathbf{BS}_{\text{FO}}$ is the set of deductively closed sets of first-order sentences,
- $\mathbf{ACC}_{\text{FO}}(\mathcal{T})$ is the set $\{ \{ \phi \in \Phi \mid \mathcal{T} \models \phi \} \}$.

Definition 7. A context $C = (L, kb, br)$ is a *first-order context* if $L = L_{\text{FO}}$, and kb is a first-order theory.

A *DL context* is similar to a first-order context but it can interpret only Description Logic ontologies (Sect. 2.1).

The *DL logic* is the logic $L_{\text{DL}} = (\mathbf{KB}_{\text{DL}}, \mathbf{BS}_{\text{DL}}, \mathbf{ACC}_{\text{DL}})$ where

- $\mathbf{KB}_{\text{DL}}$ is the set of all ontologies;
- $\mathbf{BS}_{\text{DL}}$ is the set of all deductively closed first-order theories;
- $\mathbf{ACC}_{\text{DL}}(\mathcal{O})$ is the set $\{\{ \phi \in \Phi \mid \kappa(\mathcal{O}) \models \phi \}\}$.

Definition 8. A context $C = (L, kb, br)$ is a *DL context* if $L = L_{\text{DL}}$, and kb is an ontology.

We proceed with defining an *ASP context* for logic programs (Sect. 2.2).

The *ASP logic* is the logic $L_{\text{ASP}} = (\mathbf{KB}_{\text{ASP}}, \mathbf{BS}_{\text{ASP}}, \mathbf{ACC}_{\text{ASP}})$ where

- $\mathbf{KB}_{\text{ASP}}$ is the set of all programs;
- $\mathbf{BS}_{\text{ASP}}$ is the set of all consistent subsets of $\mathbf{L}_{\mathsf{G}}$ together with $\mathbf{L}_{\mathsf{G}}$ itself;
- $\mathbf{ACC}_{\text{ASP}}(kb)$ is the set of all answer sets of kb.

Definition 9. A context $C = (L, kb, br)$ is an *ASP context* if $L = L_{\text{ASP}}$, and kb is a program $\mathcal{P}$.

A simple *database context* is used to store and retrieve facts.[4]
The *database logic* is the logic $L_{\text{DB}} = (\mathbf{KB}_{\text{DB}}, \mathbf{BS}_{\text{DB}}, \mathbf{ACC}_{\text{DB}})$ where

- $\mathbf{KB}_{\text{DB}}$ is the set of subsets of $\mathbf{L}_{\mathsf{G}}$;
- $\mathbf{BS}_{\text{DB}}$ is the set of all consistent subsets of $\mathbf{L}_{\mathsf{G}}$ together with $\mathbf{L}_{\mathsf{G}}$ itself;
- $\mathbf{ACC}_{\text{DB}}(kb)$ is the set $\{ kb \}$ if $\{ kb \}$ is consistent, and $\mathbf{L}_{\mathsf{G}}$ otherwise

Definition 10. A context $C = (L, kb, br)$ is a *database context* if $L = L_{\text{DB}}$, and kb is a subset of $\mathbf{L}_{\mathsf{G}}$.

Finally, we formally introduce MKNF logic and *MKNF contexts*.
The *MKNF logic* is the logic $L_{\text{MKNF}} = (\mathbf{KB}_{\text{MKNF}}, \mathbf{BS}_{\text{MKNF}}, \mathbf{ACC}_{\text{MKNF}})$ where

- $\mathbf{KB}_{\text{MKNF}}$ is the set of MKNF knowledge bases,
- $\mathbf{BS}_{\text{MKNF}}$ is the set of deductively closed first-order theories,
- $\mathbf{ACC}_{\text{MKNF}}(\mathcal{K})$ contains $\{ \phi \in \Phi \mid \mathcal{M} \models \phi \}$ for every MKNF model $\mathcal{M}$ of $\mathcal{K}$ and also the inconsistent belief set Φ in case $\mathcal{K}'$, obtained from $\mathcal{K}$ by removing all rules with default negation, has no MKNF model.

The latter condition is required by the formal framework of multi-context systems.

Definition 11. A context $C = (L, kb, br)$ is an *MKNF context* if $L = L_{\text{MKNF}}$, kb is an MKNF knowledge base, and, for every $\sigma \in br$, $H(\sigma)$ is an objective literal or an ontology axiom. An MKNF context $C = (L_{\text{MKNF}}, \mathcal{K}, br)$ is *ground* if $\mathcal{K}$ is ground; *finite* if both $\mathcal{K}$ and br are finite.

We can show that all these contexts are indeed reducible.

Proposition 12. *Every first-order, DL, ASP, database, and MKNF context is reducible.*

Proof. See Appendix A, page 24. □

4 Example Scenario

In this section, we describe an example MCS based on the airport scenario outlined in the introduction. The main purpose is to illustrate our approach, hence we simplify the bigger picture presented in the introduction in a number of ways: the airport MCS $M = \langle C_1, C_2, C_3, \ldots, C_n \rangle$ here comprises an ontology context C_1 with a TBox describing the relevant vocabulary of the domain and an ABox encapsulating current data, one MKNF context C_2 representing a normative monitoring system, and a number of ASP contexts (C_3 to C_n) for the cooperating security agents deployed within the airport.

In the **ontology context** we may use some readily available ontology (or possibly a combination thereof), such as the TFM ontology,[5] providing us with basic air traffic vocabulary, including the classes Airline, Flight, Passenger, Pilot, etc., but also properties Onboard, Controls, Employs, etc., expressing relations between classes. Besides such taxonomic knowledge, which is of a more static nature, the ontology context also

[4] Please note that such a database context is more expressive than a relational database and can be understood as a fact base in which we also admit negated facts, to be in line with the admitted rule language.

[5] See `http://ti.arc.nasa.gov/profile/shawn/tfmontology/` Note that we occasionally shorten/simplify some names to ease the presentation.

contains an ABox that stores current data regarding flight plans, airlines, and staff, etc. The ontology context is given as $C_1 = (L_{\text{DL}}, \mathcal{O}, \emptyset)$, a short excerpt from $\mathcal{O}$ follows:

$$\begin{array}{rlcl} \mathsf{Pilot} \sqsubseteq & \forall\,\mathsf{Controls.Aircraft} & & \mathsf{Flight(UA101)} \\ \exists\,\mathsf{Onboard}.\top \sqsubseteq & \mathsf{Passenger} & & \mathsf{Onboard(Erika, UA101)} \\ \top \sqsubseteq & \forall\,\mathsf{Onboard.Flight} & & \end{array}$$

The taxonomic part (on the left) states that all the things pilots control are aircraft, and that the relation Onboard connects passengers and flights. The data (on the right) includes information about the flight UA101, and that an individual Erika is on board of UA101. Using these axioms, it is possible to derive, e.g., that Erika is a passenger.

The **MKNF context** represents a normative monitoring system. It contains relevant norms in order to evaluate the current situation and to identify security risks in a given situation. These norms rely on expressing defaults and exceptions, so non-monotonic rules are required. The system is also equipped with sensors placed in relevant locations that are able to detect the approximate number of persons and objects present at the given location. Note that these persons and objects are not identified, so DL axioms are more suitable to represent this information. The following norms are represented: if luggage is detected at some location but no persons, then this luggage has to be inspected/removed; if there is a flight with a passenger on board with no passport, then the flight is not allowed to take off; if there is a passenger identified by a passport that is found on the blacklist, then this person/suspect has to be checked. Some specific information that is needed to implement the norms is also stored here (e.g., flights with international destinations in the IntDest/1 predicate and passport numbers that are blacklisted in the BlackList/1 predicate). This context is formally given as $C_2 = (L_{\text{MKNF}}, \mathcal{K}, br_2)$, with part of $\mathcal{K}$ and br_2 shown below:

$$\mathsf{LuggageAt} \equiv \exists\,\mathsf{Detected.Luggage} \qquad \mathsf{PersonAt} \equiv \exists\,\mathsf{Detected.Person}$$

$$\begin{aligned} \mathsf{InspectLuggageAt}(l) &\leftarrow \mathsf{Location}(l), \mathsf{LuggageAt}(l), {\sim}\mathsf{PersonAt}(l). \\ \mathsf{TakeoffNotAllowed}(f) &\leftarrow \mathsf{Flight}(f), \mathsf{IntDest}(f), \mathsf{Onboard}(x, f), {\sim}\mathsf{HasPassport}(x). \\ \mathsf{HasPassport}(x) &\leftarrow \mathsf{Passenger}(x), \mathsf{Passport}(p), \mathsf{Carries}(x, p). \\ \mathsf{CheckSuspect}(x) &\leftarrow \mathsf{Person}(x), \mathsf{Passport}(p), \mathsf{Carries}(x, p), \mathsf{BlackList}(p). \end{aligned}$$

$$\begin{array}{lll} \mathsf{Location(L1)} & \geqslant 2\,\mathsf{Detected.Person(L1)} & \leqslant 3\,\mathsf{Detected.Person(L1)} \\ \mathsf{Location(L2)} & \leqslant 5\,\mathsf{Detected.Luggage(L1)} & \geqslant 3\,\mathsf{Detected.Luggage(L2)} \end{array}$$

$$\begin{aligned} \mathsf{Flight}(f) &\leftarrow (1 : \mathsf{Flight}(f)). \\ \mathsf{Passenger}(x) &\leftarrow (1 : \mathsf{Passenger}(x)). \\ \mathsf{Passport}(p) &\leftarrow (1 : \mathsf{Passport}(p)). \\ \mathsf{Onboard}(x, f) &\leftarrow (1 : \mathsf{Onboard}(x, f)). \\ \mathsf{Carries}(x, o) &\leftarrow (1 : \mathsf{Carries}(x, o)). \end{aligned}$$

The sensor information (seen as ABox axioms) indicates that there are between 2 and 3 persons and at most 5 pieces of luggage at location L1, and at least 3 pieces of luggage at location L2. The bridge rules (at the very bottom) assure that all relevant data stored in the ontology context is imported into the normative context as well.[6]

Finally, there is a number of security agents operating autonomously but in a cooperative manner. Each agent is represented by an **agent context** where its behaviour is

[6] For the sake of concise presentation, we use variables in this section as a syntactic macro: similar to logic program rules, each bridge rule with variables represents a number of bridge rules instantiated with constants appearing in the context of origin in all possible ways.

modelled as a logic program. Recall that the whole MCS is used by one of the agents to model the complex situation at the airport. This agent (my-agent) corresponds to one of the agent contexts, the remaining agents which the my-agent reasons about are represented by the remaining agent contexts. For simplicity we use analogous programs for each agent; in reality this could be more complex (e.g. different agents would have different behaviours). We also abstract from the concrete actions but rather indicate only to investigate, and focus on showing that the agents are able to cooperate and agree on which agent will investigate what in a simplified manner. For i with $2 < i \leq n$, the agents are represented with $C_i = (L_{\text{ASP}}, \mathcal{P}, br_i)$ with $\mathcal{P}$ and br_i:

$$\begin{aligned}
\mathsf{Investigate}(l) &\leftarrow \mathsf{SuspiciousLuggage}(l), {\sim}\mathsf{UnderInvestigation}(l).\\
\mathsf{Investigate}(f) &\leftarrow \mathsf{BoardingProblem}(f), {\sim}\mathsf{UnderInvestigation}(f).\\
\mathsf{Investigate}(x) &\leftarrow \mathsf{OnBlackList}(x), {\sim}\mathsf{UnderInvestigation}(x).
\end{aligned}$$

$$\begin{aligned}
\mathsf{SuspiciousLuggage}(l) &\leftarrow (2 : \mathsf{InspectLuggageAt}(l)).\\
\mathsf{BoardingProblem}(f) &\leftarrow (2 : \mathsf{TakeoffNotAllowed}(f)).\\
\mathsf{OnBlackList}(x) &\leftarrow (2 : \mathsf{CheckSuspect}(x)).\\
\mathsf{UnderInvestigation}(z) &\leftarrow (j : \mathsf{Investigate}(z)).\ \text{for all } j \text{ with } 2 < j \leq n \text{ and } j \neq i\ .
\end{aligned}$$

The agent's program $\mathcal{P}$ together with the bridge rules indicates to investigate suspicious luggage at a location in the airport, problems related to boarding of a certain flight, or people that are identified on the airport and appear on the black list. The last bridge rule in br_i with $i > 2$ simply serves to import from all agents which situation they are currently investigating. This bridge rule (albeit in a simplistic fashion) is responsible for agent cooperation: if one agent chooses to investigate some problem, the other agents will learn this problem is already covered. Note that we abstract from further details on how agents actually negotiate who is taking care of which situation or which situation to handle first if several options exist.

We can derive that one agent will investigate location L2 because of the unattended luggage detected there. Also, if UA101 is an international flight (achieved by adding the fact IntDest(UA101) into the MKNF context) and there is no evidence in the ontology context's ABox that Erika carries a passport, then this flight is not allowed to take off, and an agent will be assigned to investigate the flight.

5 Reducing an MKNF Context

We show that MKNF knowledge bases can be used within Multi-Context Systems without using MKNF logic. First, we show that every MKNF context can be transformed into a *first-order context*. The transformed MCS has the same grounded equilibria as the original one, showing that instead of a specialised MKNF reasoner, a first-order reasoner can be used to obtain equivalent results. Then, we show that every MKNF context can be transformed into two contexts, namely a DL context and a context to store rule facts. The resulting multi-context system only requires a DL reasoner and a database (or simply a fact base) instead of an MKNF reasoner while potential non-monotonic reasoning is handled in the bridge rules. This result can be strengthened even further, resulting in a multi-context system that requires only a DL reasoner and an ASP reasoner, thus encapsulating non-monotonic bridge rules into the ASP context. This result can indeed be generalised to arbitrary MCS, ensuring that we can restrict ourselves to MCSs with purely monotonic bridge rules whenever this is beneficial.

5.1 Reduction to a First-Order Context

The transformation to a first-order context is based on transforming the rules from the MKNF knowledge base to bridge rules, leaving us with only the ontology component which can already be handled by a first-order context. For example, if the MKNF knowledge $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ in the MKNF context C_j contains the rule

$$\mathsf{InspectLuggageAt}(\mathsf{L1}) \leftarrow \mathsf{Location}(\mathsf{L1}), \mathsf{LuggageAt}(\mathsf{L1}), {\sim}\mathsf{PersonAt}(\mathsf{L1}). \ ,$$

then the corresponding first-order bridge rule is of the form:

$$\mathsf{InspectLuggageAt}(\mathsf{L1}) \leftarrow (j : \mathsf{Location}(\mathsf{L1})), (j : \mathsf{LuggageAt}(\mathsf{L1})), \mathbf{not}\,(j : \mathsf{PersonAt}(\mathsf{L1})).$$

For that purpose, we provide an abstract function that, given a program $\mathcal{P}$, provides a set of corresponding bridge rules.

Definition 13. For every integer j, $l \in \mathbf{L}_\mathsf{G}$, $B \subseteq \mathbf{L}_\mathsf{G}$, rule π and program $\mathcal{P}$ we define

$$\alpha_j(l) = (j : l), \qquad \alpha_j(B) = \{\, \alpha_j(L) \mid L \in B \,\}, \qquad \alpha_j(\mathcal{P}) = \{\, \alpha_j(\pi) \mid \pi \in \mathcal{P} \,\},$$
$$\alpha_j({\sim}l) = \mathbf{not}\,(j : l), \quad \alpha_j(\pi) = (H(\pi) \leftarrow \alpha_j(B(\pi)).).$$

Additionally, we have to consider MKNF bridge rules that have an ontology axiom ϕ in their head. This axiom needs to be translated to its first-order counterpart $\kappa(\phi)$. We simply extend κ to bridge rules.

Definition 14. For a bridge rule σ we define $\kappa(\sigma) = (\kappa(H(\sigma)) \leftarrow B(\sigma).)$ if $H(\sigma)$ is an ontology axiom and $\kappa(\sigma) = \sigma$ otherwise. For a set of bridge rules br, $\kappa(br) = \{\, \kappa(\sigma) \mid \sigma \in br \,\}$.

The definition of the first-order context that corresponds to an MKNF context is now straightforward – it suffices to apply the above two transformations α_j and κ to all rules and MKNF bridge rules of the MKNF context accordingly:

Definition 15. Let $C_j = (L_{\text{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ be a ground MKNF context. The *first-order context corresponding to* C_j is $C_j^{\text{FO}} = (L_{\text{FO}}, \{\, \kappa(\mathcal{O}) \,\}, \kappa(br_j) \cup \alpha_j(\mathcal{P}))$.

Due to the properties of the MKNF semantics, if we consider a ground MKNF context, we find that it can be substituted by the corresponding first-order context without affecting the grounded equilibria of the multi-context system. Formally:

Theorem 16. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible MCS such that for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and*

$$M' = \langle C_1, \ldots, C_{j-1}, C_j^{\text{FO}}, C_{j+1}, \ldots, C_n \rangle \ .$$

The grounded equilibria of M and M' coincide.

Proof. See Appendix A, page 27. □

If C_j is also finite,[7] then this transformation is linear: 1.) the translation $\kappa(\mathcal{O})$ is linear; 2.) each bridge rule head is translated at most once; and 3.) exactly one bridge rule is created per ground rule in $\mathcal{P}$.

This transformation can be repeated for all MKNF contexts in the multi-context system, yielding an equivalent system that does not require to use MKNF logic.

Corollary 17. *For every multi-context system M with some ground MKNF contexts, there exists a multi-context system M' such that the grounded equilibria of M and M' coincide and M' uses first-order contexts instead of the original MKNF contexts.*

[7] Finiteness is achieved in [19] by considering *DL-safe* KBs $\mathcal{K}$. Intuitively, rule applicability/grounding is restricted to constants appearing in $\mathcal{K}$. Such a notion can easily be defined for bridge rules as well.

5.2 Reduction to DL + Database

The reduction of an MKNF context C_j with $kb_j = \mathcal{O} \cup \mathcal{P}$ into a pair of contexts, a DL context and a database context, proceeds as follows: First, we transform the rules $\mathcal{P}$ from the MKNF knowledge base to bridge rules using α_j defined in Sect. 5.1. This leaves us with only the ontology component $\mathcal{O}$ and an augmented set of bridge rules. But since the literals in their heads may contain predicate symbols of arity higher than 2, which cannot be handled by ontology reasoners, we create an additional database context and redirect all bridge rules whose head literal is not used in the ontology part to this new context. The same redirection needs to be performed in the bodies of all bridge rules in the whole MCS. One side-effect is that we cannot refer to some more complex elements of the MKNF belief set directly, but, commonly, these can be expressed in a different manner. Thus, in the remainder of Sect. 5, we introduce the following restriction on MKNF contexts in MCS: for all bridge literals $(r : p)$ and $\mathbf{not}\,(r : p)$ referring to an MKNF context C_r, p is an objective literal or an ontology axiom.[8]

We formalise the embedding by first defining the division of literals appearing in the MKNF context, i.e. in the corresponding MKNF KB. Given an MKNF knowledge base $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$, we define that $l \in \mathbf{L}_\mathsf{G}$ is a *DL-literal* if the predicate symbol of l appears in $\mathcal{O}$. Otherwise, l is a *non-DL-literal*.[9]

Now we can define an abstract function that in a given MCS transforms all bridge literals that refer to some MKNF context C_j and are non-DL-literals to point to a different context k.

Definition 18. For all integers i, j, k and every $l \in \mathbf{L}_\mathsf{G}$ we define

$$\beta_j^k((i:l)) = \begin{cases} (k:l) & \text{if } i = j \text{ and } l \text{ is a non-DL-literal;} \\ (i:l) & \text{otherwise,} \end{cases} \qquad \beta_j^k(\mathbf{not}\,(i:l)) = \mathbf{not}\ \beta_j^k((i:l)).$$

Also, for a set of bridge literals B, a bridge rule σ, and a set of bridge rules br,

$$\beta_j^k(B) = \{\, \beta_j^k(L) \mid L \in B \,\},\ \beta_j^k(\sigma) = (H(\sigma) \leftarrow \beta_j^k(B(\sigma)).),\ \beta_j^k(br) = \{\, \beta_j^k(\sigma) \mid \sigma \in br \,\}.$$

Note that the second case for $\beta_j^k((i:l))$ not only handles DL-literals and ontology axioms, but also all other bridge literals not referring to the MKNF context.

We define the DL-DB MCS corresponding to an MKNF context.

Definition 19. Let $C_j = (L_{\text{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ be a ground MKNF context. The *DL-DB MCS corresponding to* C_j, $\langle C_j^{\text{DL}}, C_k^{\text{DB}} \rangle$, is defined as follows:

- $C_j^{\text{DL}} = (L_{\text{DL}}, \mathcal{O}, br_j^{\text{DL}})$, where
$$br_j^{\text{DL}} = \{\, \sigma \mid \sigma \in \beta_j^k(br_j \cup \alpha_j(\mathcal{P})) \wedge H(\sigma) \text{ is an ontology axiom} \,\} \quad ;$$
- $C_k^{\text{DB}} = (L_{\text{DB}}, \emptyset, br_k^{\text{DB}})$, where
$$br_k^{\text{DB}} = \{\, \sigma \mid \sigma \in \beta_j^k(br_j \cup \alpha_j(\mathcal{P})) \wedge H(\sigma) \text{ is a non-DL-literal} \,\} \quad .$$

Bridge rules in $\beta_j^k(br_j \cup \alpha_j(\mathcal{P}))$ are divided between the two contexts as outlined. Note that the index k for the database context allows us to add C_k^{DB} to an MCS with n contexts at a position of choice, which is simply $n+1$.

[8] Note that p is by definition an element of $\mathbf{BS}_{\text{MKNF}}$, i.e. a first-order sentence. We assume that an ontology axiom is simply a shortcut for such an element.

[9] In the case distinctions in Sect. 5, we assume that every DL-literal is also an ontology axiom.

We introduce one further notion of *closure*: given a first-order theory $\mathcal{T}$, $\mathcal{T}^*$ denotes the deductive closure of $\mathcal{T}$. The following result shows that we can substitute a finite ground MKNF context with a DL-DB MCS without affecting the grounded equilibria:

Theorem 20. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible MCS such that, for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and*

$$M' = \langle C'_1, \ldots, C'_{j-1}, C_j^{\text{DL}}, C'_{j+1}, \ldots, C'_n, C_{n+1}^{\text{DB}} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ and $i \neq j$, $C'_i = (L_i, kb_i, \beta_j^{n+1}(br_i))$.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S_j^{\text{DL}}, S'_{j+1}, \ldots, S'_n, S_{n+1}^{\text{DB}} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq j$, $S_i = S'_i$, and $S_j = (S_j^{\text{DL}} \cup S_{n+1}^{\text{DB}})^*$.

Proof. See Appendix A, page 29. □

If all contexts in M are finite, then this transformation can be done in linear time, since each rule in $\mathcal{P}$ is turned into one bridge rule, and then β_j^{n+1} is applied exactly once to each bridge rule in the MCS M.

Example 21. Recall the example scenario in Sect. 4 and assume here that $\mathcal{K}$ of C_2 as presented is complete which defines the division into DL-literals and non-DL-literals. In this case, C_2 is substituted by C_2^{DL} such that $br^{\text{DL}} = \emptyset$. Moreover, C_{n+1}^{DB} is added with br^{DB} which contains:

$$\begin{aligned}
\mathsf{InspectLuggageAt(L1)} \leftarrow\ & (2 : \mathsf{Location(L1)}), (2 : \mathsf{LuggageAt(L1)}),\\
& \mathbf{not}\,(2 : \mathsf{PersonAt(L1)}).\\
\mathsf{TakeoffNotAllowed(UA101)} \leftarrow\ & (n+1 : \mathsf{Flight(UA101)}), (2 : \mathsf{IntDest(UA101)}),\\
& (n+1 : \mathsf{Onboard(Erika, UA101)}),\\
& \mathbf{not}\,(n+1 : \mathsf{HasPassport(Erika)}).
\end{aligned}$$

Additionally, all the bridge rules of the ASP contexts referring to C_2 have to change. For example,

$$\mathsf{SuspiciousLuggage(L1)} \leftarrow (2 : \mathsf{InspectLuggageAt(L1)}).$$

becomes

$$\mathsf{SuspiciousLuggage(L1)} \leftarrow (n+1 : \mathsf{InspectLuggageAt(L1)}).$$

The transformation can be repeated for all MKNF contexts in the multi-context system, yielding an equivalent system that does not require the usage of MKNF logic.

Corollary 22. *For every multi-context system M with some ground MKNF contexts, there exists a multi-context system M' such that the grounded equilibria of M and M' are equivalent (in the sense of Theorem 20) and M' uses pairs of DL contexts and database contexts instead of the original MKNF contexts.*

5.3 Reduction to DL + ASP

The reduction of an MKNF context C_j with $kb_j = \mathcal{O} \cup \mathcal{P}$ into a pair of contexts, a DL context and an ASP context both without non-monotonic bridge rules, proceeds as follows. The ontology forms the kb of the DL context, while the non-monotonic bridge rules and $\mathcal{P}$ form the kb of the ASP context. The monotonic bridge rules are divided between the two contexts based on whether the head is an ontology axiom or a non-DL-literal as in the previous subsection. Additional monotonic bridge rules have to be added so that 1) information on DL-literals derived in the ASP context passes to the DL context, and 2) information from other contexts including the DL context is introduced to the ASP context whenever necessary.

We first provide a function that allows us to "internalise" bridge rules into the ASP context. For that purpose, new ground atoms are introduced to make sure that no complex formulas are added to the ASP context, and pieces of information from other contexts do not interfere with those already in the ASP context, i.e. the non-DL-literals.

Definition 23. For any integer i and any member p of a belief set of L_i, $\gamma_i(p)$ denotes a new ground atom uniquely associated to i and p. Also, for every integer j, and set of bridge literals B,

$$\begin{aligned} &\gamma_i((j:p)) = p \;\text{ if } i = j \;, && \gamma_i(\mathbf{not}\,(j:p)) = {\sim}\gamma_i((j:p)) \;, \\ &\gamma_i((j:p)) = \gamma_j(p) \;\text{ if } i \neq j \;, && \gamma_i(B) = \{\, \gamma_i(L) \mid L \in B \,\} \;. \end{aligned}$$

To formalise the transformation as outlined, we need some further notation. Given a set of bridge rules br, br^n is the set of all bridge rules in br that contain at least one bridge literal $\mathbf{not}\,(r:p)$ for some r and p, and $br^m = br \setminus br^n$. Also, $(i:p) \in B(\sigma)^+ \cup B(\sigma)^-$ is a short form for $((i:p) \in B(\sigma) \vee \mathbf{not}\,(i:p) \in B(\sigma))$.

We define the DL-ASP MCS corresponding to an MKNF context as follows. We apply β_j^k to the monotonic bridge rules to set the pointers correctly w.r.t. the two new contexts, obtaining the set of bridge rules br'_j, which is then divided as bridge rules between the two contexts. Furthermore, we translate $\mathcal{P}$ into bridge rules using α_j and join these to the non-monotonic bridge rules in the MKNF context, and then we also apply β_j^k to the result. This results in the set of bridge rules br''_j which provides the kb of the ASP context after a suitable translation into logic programing rules using the function γ defined above. Additionally, auxiliary bridge rules are added to the DL context to introduce information from non-monotonic rules with ontology axioms as heads. Likewise, auxiliary bridge rules are added to the ASP context to introduce information from the DL context as well as other contexts in the MCS.

Definition 24. Let $C_j = (L_{\mathrm{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ be a ground MKNF context, k an integer, $br'_j = \beta_j^k(br_j^m)$ and $br''_j = \beta_j^k(br_j^n \cup \alpha_j(\mathcal{P}))$. The *DL-ASP MCS corresponding to* C_j, $\langle C_j^{\mathrm{DL}_m}, C_k^{\mathrm{ASP}} \rangle$, is defined as follows:

- $C_j^{\mathrm{DL}_m} = (L_{\mathrm{DL}}, \mathcal{O}, br_j^{\mathrm{DL}_m})$, where

$$\begin{aligned} br_j^{\mathrm{DL}_m} = {} & \{\, \sigma \mid \sigma \in br'_j \wedge H(\sigma) \text{ is an ontology axiom} \,\} \\ & \cup \{\, H(\sigma) \leftarrow (k : \gamma_j(H(\sigma))). \mid \sigma \in br''_j \wedge H(\sigma) \text{ is an ontology axiom} \,\} \;; \end{aligned}$$

- $C_k^{\mathrm{ASP}} = (L_{\mathrm{ASP}}, kb_k, br_k^{\mathrm{ASP}})$, where

$$\begin{aligned} kb_k = {} & \{\, \gamma_j(H(\sigma)) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in br''_j \wedge H(\sigma) \text{ is an ontology axiom} \,\} \\ & \cup \{\, H(\sigma) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in br''_j \wedge H(\sigma) \text{ is a non-DL-literal} \,\} \;, \end{aligned}$$

$$\begin{aligned} br_k^{\mathrm{ASP}} = {} & \{\, \sigma \mid \sigma \in br'_j \wedge H(\sigma) \text{ is a non-DL-literal} \,\} \\ & \cup \{\, \gamma_i(p) \leftarrow (i:p). \mid \sigma \in br''_j \wedge k \neq i \wedge (i:p) \in B(\sigma)^+ \cup B(\sigma)^- \,\} \;. \end{aligned}$$

The application of γ and the case distinction for kb_k make sure that everything in kb_k but the non-DL-literals from the MKNF context are newly introduced atoms.

We can substitute a ground MKNF context with a DL-ASP MCS without affecting the grounded equilibria.

Theorem 25. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible multi-context system such that, for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and*

$$M' = \langle C'_1, \ldots, C'_{j-1}, C_j^{\mathrm{DL}_m}, C'_{j+1}, \ldots, C'_n, C_{n+1}^{\mathrm{ASP}} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ and $i \neq j$, $C'_i = (L_i, kb_i, \beta_j^{n+1}(br_i))$. The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S_j^{\mathrm{DL}_m}, S'_{j+1}, \ldots, S'_n, S_{n+1}^{\mathrm{ASP}} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \neq j \leq n$, $S_i = S'_i$, and $S_j = (S_j^{\mathrm{DL}_m} \cup (S_{n+1}^{\mathrm{ASP}} \setminus S^{\mathrm{AUX}}))^*$ where $S^{\mathrm{AUX}} \subseteq S_{n+1}^{\mathrm{ASP}}$ is the set of all new ground atoms introduced by γ.

Proof. See Appendix A, page 31. □

Example 26. Consider the two non-monotonic bridge rules in the database context in Example 21. Both become rules of $\mathcal{P}$ in the ASP context.

$$\begin{aligned}
\mathsf{InspectLuggageAt(L1)} &\leftarrow \mathsf{Location(L1), LuggageAt(L1), {\sim}PersonAt(L1)}.\\
\mathsf{TakeoffNotAllowed(UA101)} &\leftarrow \mathsf{Flight(UA101), IntDest(UA101),}\\
&\qquad \mathsf{Onboard(Erika, UA101), {\sim}HasPassport(Erika)}.
\end{aligned}$$

Additionally, we need to add bridge rules to import information from the DL context for the three atoms in the first rule, such as:

$$\mathsf{Location(L1)} \leftarrow (2 : \mathsf{Location(L1)}).$$

This transformation can be repeated for all MKNF contexts in the multi-context system, yielding an equivalent system that does neither require the usage of MKNF logic nor non-monotonic bridge rules in their substitute contexts. Formally:

Corollary 27. *For every multi-context system M with some ground MKNF contexts, there exists a multi-context system M' such that the grounded equilibria of M and M' are equivalent (in the sense of Theorem 25) and M' uses pairs of DL contexts and ASP contexts with monotonic bridge rules instead of the original MKNF contexts.*

Furthermore, the transformation can be generalised even further: whenever an ASP context is present in (or added to) the MCS in consideration, then all non-monotonic bridge rules can be transformed away into that context. Indeed, non-monotonic bridge rules are simply all transferred into the ASP context. Each such transferred bridge rule is substituted by a monotonic one linking the information back from the ASP context to the context of origin. The ASP context is augmented with additional bridge rules that add information requested from other contexts in the former bridge literals.

Theorem 28. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible MCS with $C_i = (L_i, kb_i, br_i)$ for all i with $1 \leq i \leq n$ and C_j a ground ASP context for some j with $1 \leq j \leq n$, and $M' = \langle C'_1, \ldots, C'_n \rangle$ with*

- *$C'_i = (L_i, kb_i, br'_i)$ for all i with and $i \neq j$, where*

  $$br'_i = br_i^m \cup \{\, H(\sigma) \leftarrow (j : \gamma_i(H(\sigma))). \mid \sigma \in br_i^n \,\} \quad ;$$

- *$C'_j = (L_j, kb'_j, br'_j)$, where $kb'_j = kb_j \cup \bigcup_i \{\gamma_i(H(\sigma)) \leftarrow \gamma_j(B(\sigma)). \mid \sigma \in br_i^n\}$, and*

  $$br'_j = br_j^m \cup \textstyle\bigcup_i \{\, \gamma_k(p) \leftarrow (k : p). \mid \sigma \in br_i^n \wedge j \neq k \wedge (k : p) \in B(\sigma)^+ \cup B(\sigma)^- \,\}\,.$$

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \dots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \dots, S'_n \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq j$, $S_i = S'_i$, and $S_j = S'_j \setminus S^{\text{AUX}}$ where $S^{\text{AUX}} \subseteq S'_j$ is the set of all new ground atoms introduced by γ.

Proof. See Appendix A, page 31. □

Note that, since we can simply introduce a new ASP context, the result is indeed general.

Finally, assuming that we measure the size of the MCS in terms of the size of its (finite) contexts as well as of bridge rules (number of bridge literals), then this transformation is linear as it produces, in the ASP context, at most one new bridge rule per bridge literal (both for Theorem 25 and 28).

6 Reducing to MKNF Contexts

To complete the overall picture, we also provide a transformation in the opposite direction, i.e. we take a set of DL, DB, ASP and MKNF contexts and transform them into a single MKNF context. In light of the results in the previous section, and to simplify the presentation, w.l.o.g. we assume that sets of predicate symbols used in different contexts are disjoint.[10]

We can now consider joining two MKNF contexts into one. Before we define the reduction, we need an additional function that re-sets the contexts to which bridge literals point to account for the removal of one context from the MCS.

Definition 29. For all integers i, j, k with $j < k$ and every $l \in \mathbf{L}_{\mathsf{G}}$ we define

$$\delta_j^k((i:l)) = \begin{cases} (i:l) & \text{if } i < k; \\ (j:l) & \text{if } i = k; \\ ((i-1):l) & \text{otherwise.} \end{cases} \qquad \delta_j^k(\mathbf{not}\,(i:l)) = \mathbf{not}\ \delta_j^k((i:l)).$$

Also, for a set of bridge literals B, a bridge rule σ, and a set of bridge rules br,

$$\delta_j^k(B) = \{\, \delta_j^k(L) \mid L \in B \,\},\ \delta_j^k(\sigma) = (H(\sigma) \leftarrow \delta_j^k(B(\sigma)).),\ \delta_j^k(br) = \{\, \delta_j^k(\sigma) \mid \sigma \in br \,\}.$$

We define a joint MKNF context as follows.

Definition 30. Let $C_j^1 = (L_{\text{MKNF}}, \mathcal{O}^1 \cup \mathcal{P}^1, br_j^1)$ and $C_k^2 = (L_{\text{MKNF}}, \mathcal{O}^2 \cup \mathcal{P}^2, br_k^2)$ be MKNF contexts. The *MKNF context corresponding to C_j^1 and C_k^2*, C_j^{MKNF} is defined as $C_j^{\text{MKNF}} = (L_{\text{MKNF}}, (\mathcal{O}^1 \cup \mathcal{O}^2) \cup (\mathcal{P}^1 \cup \mathcal{P}^2), \delta_j^k(br_j^1 \cup br_k^2))$.

We can transform an MCS with two MKNF contexts into one with only one context.

Theorem 31. *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible multi-context system such that, for some j and k with $1 \leq j < k \leq n$, C_j and C_k are ground MKNF contexts, and*

$$M' = \langle C'_1, \dots, C'_{j-1}, C_j^{\text{MKNF}}, C'_{j+1}, \dots, C'_{n-1} \rangle$$

where, given $C_i = (L_i, kb_i, br_i)$, for all i with $1 \leq i < n$ and $i \neq j$, $C'_i = (L_i, kb_i, \delta_j^k(br_i))$ if $i < k$, and $C'_i = (L_{i+1}, kb_{i+1}, \delta_j^k(br_{i+1}))$ otherwise.

[10] Note that from the point of view of MCSs, predicate symbols with the same name are interpreted independently of one another when used in different contexts and must be interconnected using bridge rules if their interpretation needs to be unified across contexts. The disjointness of predicate names used in different contexts can thus be achieved by a simple renaming transformation that does not affect the semantics of the MCS.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S_j^{\text{MKNF}}, S'_{j+1}, \ldots, S'_{n-1} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i < n$ and $i \neq j$, $S'_i = S_i$ if $i < k$ and $S'_i = S_{i+1}$ otherwise, and $S_j^{\text{MKNF}} = (S_j \cup S_k)^*$.

Proof. See Appendix A, page 32. □

This transformation is again linear for finite ground MKNF contexts, since two contexts are simply joined only re-directing some context references.

We can generalise to an arbitrary number of MKNF, DL, ASP, and DB contexts.

Corollary 32. *For every multi-context system M with some ground MKNF, ASP, and DL contexts, there exists a multi-context system M' such that the grounded equilibria of M and M' are equivalent (in the sense of applying Theorem 31 stepwise) and M' uses only one MKNF context instead of the original MKNF, ASP, and DL contexts.*

Note that, if all contexts in a given MCS are DL, DB, ASP, and MKNF contexts, then we can obtain a hybrid MKNF knowledge base.

7 Conclusions

We have shown that MKNF knowledge bases can be used in MCSs as one context, thereby enabling us to use these highly expressive knowledge representation and reasoning formalisms in such MCSs. MKNF contexts can then be interlinked with other contexts in the MCS yielding an even richer formalism.

Since there may be agents that are not capable of reasoning with such a complex context, we investigated the potential usage of MKNF knowledge bases without requiring the presence of an actual MKNF context or an adequate reasoning service. We have shown that we can use a first-order context instead where the rule part of the MKNF context is transformed into bridge rules. Alternatively, we can use a DL context and a simple database context, both with non-monotonic bridge rules, to represent an MKNF context. Moreover, we can rely on a DL context and an ASP context, both with monotonic bridge rules only, to emulate an MKNF context, but encapsulating the non-monotonic reasoning steps into one context which may be handled in an external answer set solver. This result can even be generalised to the entire MCS, resulting in an MCS where all non-monotonic bridge rules are transferred into the ASP context. Finally, we also provided a transformation for the inverse direction, i.e. if we have the capability of an MKNF reasoner, then we can merge all DL, database, ASP, and MKNF contexts into one MKNF context only.

Future work may extend the results to more general MKNF KBs and likewise more complex MKNF contexts, i.e. where literals in MKNF rules and bridge literals referring to an MKNF context may be extended to more complex formulas beyond ontology axioms and objective literals in bridge rule heads. In [6], also a well-founded semantics is defined for MCSs and considering this semantics and investigating its correlation with the well-founded semantics for Hybrid MKNF [17] would also be interesting, possibly enabling us to use a semantics in MCSs that is, due to its nature, of a lower computational complexity. Finally, we may consider an extension to managed Multi-Context Systems [7] in which bridge rules can not only add information to other contexts but are more general, e.g. deletion or revision operators can be defined.

Acknowledgments Matthias Knorr and João Leite were partially supported by Fundação para a Ciência e a Tecnologia under project "ERRO – Efficient Reasoning with Rules and Ontologies" (PTDC/EIA-CCO/121823/2010) and Matthias Knorr also by FCT grant SFRH/BPD/86970/2012. Martin Slota was partially supported by Fundação para a Ciência e a Tecnologia under project "ASPEN – Answer Set Programming with BoolEaN Satisfiability" (PTDC/EIA-CCO/110921/2009). Martin Homola was partially supported by projects "Knowledge Representation for Ambient Intelligence" (VEGA project no. 1/1333/12) and "Measuring, communication and information systems for monitoring of cardiovascular risk in hypertension patients" (APVV project no. APVV-0513-10). The collaboration between the co-authors resulted from the Slovak–Portuguese bilateral project "ReDIK – Reasoning with Dynamic Inconsistent Knowledge", supported by APVV agency under SK-PT-0028-10 and by Fundação para a Ciência e a Tecnologia (FCT/2487/3/6/2011/S).

References

[1] M. Alberti, A. S. Gomes, R. Gonçalves, M. Knorr, J. Leite, and M. Slota. Normative systems require hybrid knowledge bases (extended abstract). In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Valencia, Spain, June 4-8 2012. IFAAMAS.

[2] M. Alberti, A. S. Gomes, R. Gonçalves, J. Leite, and M. Slota. Normative systems represented as hybrid knowledge bases. In *Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17-18, 2011. Proceedings*, volume 6814 of *Lecture Notes in Computer Science*, pages 330–346. Springer, 2011.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition, 2007.

[4] M. Benerecetti, A. Cimatti, E. Giunchiglia, F. Giunchiglia, and L. Serafini. Formal specification of beliefs in multi-agent systems. In *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*, volume 1193 of *Lecture Notes in Computer Science*, pages 117–130. Springer, 1997.

[5] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model checking multiagent systems. *Journal of Logic and Computation*, 8(3):401–423, 1998.

[6] G. Brewka and T. Eiter. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 385–390, Vancouver, British Columbia, Canada, July 22-26 2007. AAAI Press.

[7] G. Brewka, T. Eiter, M. Fink, and A. Weinzierl. Managed multi-context systems. In T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 786–791. IJCAI/AAAI, 2011.

[8] A. Casali, L. Godo, and C. Sierra. Graded BDI models for agent architectures. In *Computational Logic in Multi-Agent Systems, 5th International Workshop,*

CLIMA V, Lisbon, Portugal, September 29-30, 2004, Revised Selected and Invited Papers, volume 3487 of *Lecture Notes in Computer Science*, pages 126–143. Springer, 2005.

[9] A. Cimatti and L. Serafini. Multi-agent reasoning with belief contexts: The approach and a case study. In *Intelligent Agents, ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Amsterdam, The Netherlands, August 8-9, 1994, Proceedings*, volume 890 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 1995.

[10] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Graduate texts in computer science. Springer-Verlag, Berlin, Germany, 2nd edition, 1996.

[11] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-driven answer set solving. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 386–392. AAAI Press/The MIT Press, 2007.

[12] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3-4):365–385, 1991.

[13] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning=locality+compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[14] F. Giunchiglia. Contextual reasoning. *Epistemologia - Special Issue on I Linguaggi e le Macchine*, XVI:345–364, 1993.

[15] A. S. Gomes, J. J. Alferes, and T. Swift. Implementing query answering for hybrid MKNF knowledge bases. In *Proceedings of the 12th International Symposium on Practical Aspects of Declarative Languages (PADL 2010)*, volume 5937 of *Lecture Notes in Computer Science*, pages 25–39, Madrid, Spain, January 18-19 2010. Springer.

[16] V. Ivanov, M. Knorr, and J. Leite. A query tool for $\mathcal{EL}$ with non-monotonic rules. In *ISWC 2013*. Springer, 2013. to appear.

[17] M. Knorr, J. J. Alferes, and P. Hitzler:. Local closed world reasoning with description logics under the well-founded semantics. *Artificial Intelligence*, 175(9-10):1528–1554, 2011.

[18] V. Lifschitz. Nonmonotonic databases and epistemic queries. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91). Sydney, Australia, August 24-30, 1991*, pages 381–386. Morgan Kaufmann, 1991.

[19] B. Motik and R. Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):93–154, 2010.

[20] B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.

[21] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[22] P. F. Patel-Schneider and I. Horrocks. A comparison of two modelling paradigms in the semantic web. *J. Web Sem.*, 5(4):240–250, 2007.

[23] J. Sabater, C. Sierra, S. Parsons, and N. R. Jennings. Engineering executable agents using multi-context systems. *Journal of Logic and Computation*, 12(3):413–442, 2002.

A Proofs

A.1 Basic Properties of MKNF

Similarly as shown in [19], MKNF knowledge bases are faithful to both the semantics of ontologies and to the answer set semantics of logic programs. In the case of ontologies, this follows from [10, Theorem 5.9.4] and the fact that we work under the UNA, so equality is interpreted the same way for ontologies and MKNF theories.

Proposition 33. *Let $\mathcal{O}$ be an ontology, $\mathcal{K}$ the MKNF knowledge base $\mathcal{O} \cup \emptyset$ and ϕ a first-order sentence. Then $\mathcal{O} \models \phi$ if and only if for every MKNF model $\mathcal{M}$ of $\mathcal{K}$, $\mathcal{M} \models \phi$.*

Furthermore, a logic program is just a special case of an MKNF knowledge base, and the unique name assumption imposed on MKNF interpretations is sufficient to conclude the following:

Proposition 34. *Let $\mathcal{P}$ be a logic program and $\mathcal{K}$ be the MKNF knowledge base $\emptyset \cup \mathcal{P}$. If $J \subseteq \mathbf{L}_\mathsf{G}$ is a consistent answer set of $\mathcal{P}$, then $\{ I \in \mathbf{I} \mid I \models J \}$ is an MKNF model of $\mathcal{K}$. If $\mathcal{M}$ is an MKNF model $\mathcal{K}$, then $\{ l \in \mathbf{L}_\mathsf{G} \mid \mathcal{M} \models l \}$ is an answer set of $\mathcal{P}$.*

In other words, the consistent answer sets of a logic program $\mathcal{P}$ are in one-to-one correspondence with MKNF models of the MKNF knowledge base $(\emptyset, \mathcal{P})$.

In the remainder, we will focus only on *ground* MKNF knowledge bases. This simplification is justified by the following lemma which easily follows from the UNA satisfied by MKNF interpretations.

Lemma 35. *Any MKNF knowledge base has the same MKNF models as its grounding.*

A.2 Properties of MKNF Models

Definition 36. An MKNF formula ϕ is *subjective* if every first-order atom in ϕ is in the scope of a modal operator. That is, ϕ is of one of the following forms: $\mathbf{K}\,\psi$, $\mathbf{not}\,\psi$, $\neg\phi_1$, $\phi_1 \wedge \phi_2$ where ψ is an MKNF formula and ϕ_1, ϕ_2 are subjective MKNF formulas.

Lemma 37. *Let ϕ be a subjective MKNF formula, $I \in \mathbf{I}$ and $\mathcal{M}$ an MKNF interpretation. Then, $(I, \mathcal{M}, \mathcal{M}) \models \phi$ if and only if $\mathcal{M} \models \phi$.*

Proof. We consider the different forms that ϕ can take separately:

a) If $\phi = \mathbf{K}\,\psi$, then $(I, \mathcal{M}, \mathcal{M}) \models \phi$ if and only if $(J, \mathcal{M}, \mathcal{M}) \models \psi$ for all $J \in \mathcal{M}$, which is equivalent to $(J, \mathcal{M}, \mathcal{M}) \models \mathbf{K}\,\psi$ for all $J \in \mathcal{M}$, which in turn is the same as $\mathcal{M} \models \phi$.

b) If $\phi = \mathbf{not}\,\psi$, then $(I, \mathcal{M}, \mathcal{M}) \models \phi$ if and only if $(J, \mathcal{M}, \mathcal{M}) \not\models \psi$ for some $J \in \mathcal{M}$, which, since $\mathcal{M}$ is non-empty, is equivalent to $(J, \mathcal{M}, \mathcal{M}) \models \mathbf{not}\,\psi$ for all $J \in \mathcal{M}$, which in turn is the same as $\mathcal{M} \models \phi$.

c) If $\phi = \neg\phi_1$, then we can inductively assume that the claim holds for ϕ_1 and we obtain $(I, \mathcal{M}, \mathcal{M}) \models \phi$ if and only if $(I, \mathcal{M}, \mathcal{M}) \not\models \phi_1$, by the hypothesis this is equivalent to $\mathcal{M} \not\models \phi_1$ and since $\mathcal{M}$ is non-empty, using the hypothesis again we can rewrite this as $(J, \mathcal{M}, \mathcal{M}) \not\models \phi_1$ for all $J \in \mathcal{M}$, which in turn is the same as $(J, \mathcal{M}, \mathcal{M}) \models \phi$ for all $J \in \mathcal{M}$, or in other words $\mathcal{M} \models \phi$.

d) If $\phi = \phi_1 \wedge \phi_2$, then we can inductively assume that the claim holds for ϕ_1 and ϕ_2 and we obtain $(I, \mathcal{M}, \mathcal{M}) \models \phi$ if and only if both $(I, \mathcal{M}, \mathcal{M}) \models \phi_1$ and $(I, \mathcal{M}, \mathcal{M}) \models \phi_2$, by the hypothesis this is equivalent to $\mathcal{M} \models \phi_1$ and $\mathcal{M} \models \phi_2$, rewritable as $(J, \mathcal{M}, \mathcal{M}) \models \phi_1$ and $(J, \mathcal{M}, \mathcal{M}) \models \phi_2$ for all $J \in \mathcal{M}$, which in turn is equivalent to $(J, \mathcal{M}, \mathcal{M}) \models \phi$ for all $J \in \mathcal{M}$, in other words $\mathcal{M} \models \phi$. □

According to the following Lemma, for any rule π, in order to prove $\mathcal{M} \models \kappa(\pi)$, it suffices to prove that $\mathcal{M} \models \kappa(B(\pi))$ implies $\mathcal{M} \models \kappa(H(\pi))$.

Lemma 38. *Let ϕ be a subjective MKNF formula of the form $\phi_1 \supset \phi_2$ and $\mathcal{M}$ an MKNF interpretation. Then $\mathcal{M} \models \phi$ if and only if $\mathcal{M} \models \phi_1$ implies $\mathcal{M} \models \phi_2$.*

Proof. Follows directly from Lemma 37 applied to ϕ_1 and ϕ_2. □

Proposition 39. *Every definite MKNF knowledge base either has no S5 model, or it has a unique MKNF model which coincides with its greatest S5 model.*

Proof. Suppose that the MKNF knowledge base $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ has some S5 model and let $\mathcal{M}$ be the union of all S5 models of $\mathcal{K}$. First we show that $\mathcal{M}$ is an S5 model of $\mathcal{K}$, i.e. it is the greatest S5 model of $\mathcal{M}$.

Take some MKNF formula ϕ from $\mathcal{K}$. If $\phi = \mathbf{K}\, \kappa(\mathcal{O})$, then we need to show that for all $I \in \mathcal{M}$, $(I, \mathcal{M}, \mathcal{M}) \models \kappa(\mathcal{O})$. Since $\kappa(\mathcal{O})$ is a first-order sentence, this holds if and only if for all $I \in \mathcal{M}$, $I \models \kappa(\mathcal{O})$. Take some $I \in \mathcal{M}$. Then I belongs to some S5 model $\mathcal{N}$ of $\mathcal{K}$ and, consequently, $I \models \kappa(\mathcal{O})$.

The other possibility is that ϕ is a formula of the form $\kappa(B(\pi)) \supset \kappa(H(\pi))$ for some $\pi \in \mathcal{P}$. We need to show that $\mathcal{M} \models \phi$. Suppose that $\mathcal{M} \models \kappa(B(\pi))$, we will prove that $\mathcal{M} \models \kappa(H(\pi)) = \mathbf{K}\, H(\pi)$. Take an arbitrary $J \in \mathcal{M}$ and some S5 model $\mathcal{N}$ of $\mathcal{K}$ such that $J \in \mathcal{N}$. Since $B(\pi)$ does not contain default literals, it follows from $\mathcal{M} \models \kappa(B(\pi))$ and $\mathcal{N} \subseteq \mathcal{M}$ that $\mathcal{N} \models \kappa(B(\pi))$ and thus it must be the case that $J \models H(\pi)$. Consequently, we can conclude that $\mathcal{M} \models \mathbf{K}\, H(\pi)$.

It remains to show that $\mathcal{M}$ is the unique MKNF model of $\mathcal{K}$. Since $\kappa(\mathcal{K})$ does not contain $\mathbf{not}$, it follows by the definitions of MKNF satisfaction and of an MKNF model that the MKNF models of $\mathcal{K}$ are exactly its subset-maximal S5 models. Since $\mathcal{M}$ is the greatest S5 model of $\mathcal{M}$, it follows that it is also its unique MKNF model. □

Lemma 40. *Let $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ be a definite MKNF knowledge base. If $\mathcal{K}$ has no MKNF model, then $\mathcal{O} \cup \{\, (H(\pi).) \mid \pi \in \mathcal{P} \,\}$ has no MKNF model either.*

Proof. If $\mathcal{O} \cup \{\, (H(\pi).) \mid \pi \in \mathcal{P} \,\}$ has an MKNF model, then it has an S5 model which must also be an S5 model of $\mathcal{K}$. Thus it follows from Proposition 39 that $\mathcal{K}$ has an MKNF model, too. □

Lemma 41. *Let $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ be a definite MKNF knowledge base, $\mathcal{M}$ an MKNF interpretation and $\mathcal{K}' = \mathcal{O} \cup \{\, (H(\pi).) \mid \pi \in \mathcal{P} \wedge \mathcal{M} \models \kappa(B(\pi)) \,\}$. $\mathcal{M}$ is an S5 model of $\mathcal{K}$ if and only if $\mathcal{M}$ is an S5 model of $\mathcal{K}'$. Also, if $\mathcal{M}$ is the MKNF model of $\mathcal{K}$, then $\mathcal{M}$ is the MKNF model of $\mathcal{K}'$.*

Proof. Both claims follow from Lemma 38 and from basic properties of MKNF interpretations. □

Definition 42. Let $\mathcal{K} = \mathcal{O} \cup \mathcal{P}$ be an MKNF knowledge base and $\mathcal{M}$ an MKNF interpretation. The *reduct of* $\mathcal{K}$ *relative to* $\mathcal{M}$ is the MKNF knowledge base $\mathcal{K}^{\mathcal{M}} = \mathcal{O} \cup \mathcal{P}^{\mathcal{M}}$ where $\mathcal{P}^{\mathcal{M}} = \{\, H(\pi) \leftarrow B(\pi)^{+}. \mid \pi \in \mathcal{P} \wedge \mathcal{M} \models \kappa({\sim}B(\pi)^{-}) \,\}$.

Lemma 43. *Let $\mathcal{K}$ be an MKNF knowledge base and $\mathcal{M}$ an MKNF interpretation. Then $\mathcal{M}$ is an S5 model of $\mathcal{K}$ if and only if $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$.*

Proof. Suppose that $\mathcal{M}$ is an S5 model of $\mathcal{K}$. Obviously, $\mathcal{M} \models \kappa(\mathcal{O})$. Take some $\pi' = (H(\pi) \leftarrow B(\pi)^{+}.)$ from $\mathcal{P}^{\mathcal{M}}$ for some $\pi \in \mathcal{P}$ with $\mathcal{M} \models \kappa({\sim}B(\pi)^{-})$. Then $\kappa(\mathcal{K}^{\mathcal{M}})$ contains the formula $\kappa(\pi')$ of the form $\kappa(B(\pi)^{+}) \supset \kappa(H(\pi))$. If $\mathcal{M} \models \kappa(B(\pi)^{+})$, then it is easy to verify that $\mathcal{M} \models \kappa(B(\pi))$ and thus since $\mathcal{M} \models \kappa(\pi)$, it follows that $\mathcal{M} \models \kappa(H(\pi))$. Hence, $\mathcal{M} \models \kappa(\pi')$.

For the converse implication, assume that $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$. Obviously, $\mathcal{M} \models \kappa(\mathcal{O})$, so consider some rule $\pi \in \mathcal{P}$. If $\mathcal{M} \models \kappa(B(\pi))$, then $\mathcal{P}^{\mathcal{M}}$ contains the rule $\pi' = (H(\pi) \leftarrow B(\pi)^{+}.)$ and since $\mathcal{M} \models \kappa(\pi')$, it follows that $\mathcal{M} \models \kappa(H(\pi))$. Thus, $\mathcal{M} \models \kappa(\pi)$. □

Proposition 44. *An MKNF interpretation $\mathcal{M}$ is an MKNF model of an MKNF knowledge base $\mathcal{K}$ if and only if $\mathcal{M}$ is an MKNF model of $\mathcal{K}^{\mathcal{M}}$.*

Proof. Suppose that $\mathcal{M}$ is an MKNF model of $\mathcal{K}$. Then it is also an S5 model of $\mathcal{K}$, so it follows that it is an S5 model of $\mathcal{K}^{\mathcal{M}}$ from Lemma 43.

Since $\mathcal{M}$ is an S5 model of $\mathcal{K}^{\mathcal{M}}$, it must hold that $\mathcal{M}$ is a subset of the greatest S5 model $\mathcal{M}'$ of $\mathcal{K}^{\mathcal{M}}$. By contradiction, we will show that $\mathcal{M} = \mathcal{M}'$, i.e. $\mathcal{M}$ is the MKNF model of $\mathcal{K}^{\mathcal{M}}$ (c.f. Proposition 39). Assume $\mathcal{M} \subsetneq \mathcal{M}'$. Since $\mathcal{M}$ is an MKNF model of $\mathcal{K}$, there must be some formula $\phi \in \kappa(\mathcal{K})$ and some $I' \in \mathcal{M}'$ such that $(I', \mathcal{M}', \mathcal{M}) \not\models \phi$. But $\mathcal{M}' \models \kappa(\mathcal{O})$, so ϕ must be of the form $\kappa(B(\pi)) \supset \kappa(H(\pi))$ for some rule $\pi \in \mathcal{P}$ and the following must hold:

$$(I', \mathcal{M}', \mathcal{M}) \models \kappa({\sim}B(\pi)^{-}) \wedge (I', \mathcal{M}', \mathcal{M}) \models \kappa(B(\pi)^{+}) \wedge (I', \mathcal{M}', \mathcal{M}) \not\models \kappa(H(\pi))$$

which is equivalent to

$$\mathcal{M} \models \kappa({\sim}B(\pi)^{-}) \wedge \mathcal{M}' \models \kappa(B(\pi)^{+}) \wedge \mathcal{M}' \not\models \kappa(H(\pi)) \enspace .$$

However, this is in conflict with $\mathcal{M}'$ being an S5 model of $\mathcal{K}^{\mathcal{M}}$ since the formula $\kappa(B(\pi)^{+}) \supset \kappa(H(\pi))$ belongs to $\kappa(\mathcal{K}^{\mathcal{M}})$.

For the converse implication, assume that $\mathcal{M}$ is the MKNF model of $\mathcal{K}^{\mathcal{M}}$. Then it follows from Lemma 43 that $\mathcal{M}$ is an S5 model of $\mathcal{K}$.

Take some $\mathcal{M}' \supsetneq \mathcal{M}$. Since $\mathcal{M}$ is the greatest S5 model of $\mathcal{K}^{\mathcal{M}}$, there is some rule $\pi' = (H(\pi) \leftarrow B(\pi)^{+}.) \in \mathcal{P}^{\mathcal{M}}$ such that $\mathcal{M}' \not\models \kappa(\pi')$, i.e.

$$\mathcal{M} \models \kappa({\sim}B(\pi)^{-}) \wedge \mathcal{M}' \models \kappa(B(\pi)^{+}) \wedge \mathcal{M}' \not\models \kappa(H(\pi))$$

For any $I' \in \mathcal{M}'$, this is equivalent to

$$(I', \mathcal{M}', \mathcal{M}) \models \kappa({\sim}B(\pi)^{-}) \wedge (I', \mathcal{M}', \mathcal{M}) \models \kappa(B(\pi)^{+}) \wedge (I', \mathcal{M}', \mathcal{M}) \not\models \kappa(H(\pi))$$

which is equivalent to $(I', \mathcal{M}', \mathcal{M}) \not\models \kappa(\pi)$. Thus, $\mathcal{M}$ is an MKNF model of $\mathcal{K}$. □

A.3 Contexts

Proposition 12. *Every first-order, DL, ASP, database, and MKNF context is reducible.*

Proof. We show the argument for MKNF contexts, all other cases are similar.

First we need to prove that the MKNF logic L_{MKNF} is reducible. Let $\mathbf{KB}^*_{\text{MKNF}}$ be the set of definite MKNF knowledge bases. It follows from Proposition 39 that the reduction of $\mathbf{ACC}_{\text{MKNF}}$ to $\mathbf{KB}^*_{\text{MKNF}}$ is a monotonic logic.

Let the reduction function $\mathit{red}_{\text{MKNF}} : \mathbf{KB}_{\text{MKNF}} \times \mathbf{BS}_{\text{MKNF}} \to \mathbf{KB}^*_{\text{MKNF}}$ be defined as follows:

$$\mathit{red}_{\text{MKNF}}(\mathcal{O} \cup \mathcal{P}, S) = \mathcal{O} \cup \mathcal{P}^S$$

where $\mathcal{P}^S = \{ H(\pi) \leftarrow B(\pi)^+. \mid \pi \in \mathcal{P} \wedge B(\pi)^- \cap S = \emptyset \}$. We need to verify the following conditions:

1. $\mathit{red}_{\text{MKNF}}(\mathcal{K}, S) = \mathcal{K}$ whenever $\mathcal{K} \in \mathbf{KB}^*_{\text{MKNF}}$,
2. $\mathit{red}_{\text{MKNF}}(\mathcal{K}, S) \subseteq \mathit{red}_{\text{MKNF}}(\mathcal{K}, S')$ whenever $S' \subseteq S$,
3. $S \in \mathbf{ACC}_{\text{MKNF}}(\mathcal{K})$ if and only if $\mathbf{ACC}_{\text{MKNF}}(\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)) = \{ S \}$.

The first condition follows from the fact that $\mathcal{P}^S = \mathcal{P}$ when $\mathcal{K}$ is definite.

To see that the second condition is satisfied, suppose that $S' \subseteq S$. If $\mathcal{P}^S$ contains the rule $\pi' = (H(\pi) \leftarrow B(\pi)^+.)$ for some $\pi \in \mathcal{P}$, then $B(\pi)^- \cap S = \emptyset$, and from $S' \subseteq S$ we obtain $B(\pi)^- \cap S' = \emptyset$. Thus, π' belongs to $\mathcal{P}^{S'}$ as well.

To verify the final condition, first suppose that $\Phi \in \mathbf{ACC}_{\text{MKNF}}(\mathcal{K})$. Then $\mathcal{K}'$, obtained from $\mathcal{K}$ by removing all rules with default negation, has no MKNF model. Also, $\mathcal{K}'$ coincides with $\mathit{red}_{\text{MKNF}}(\mathcal{K}, \Phi)$, so $\mathbf{ACC}_{\text{MKNF}}(\mathit{red}_{\text{MKNF}}(\mathcal{K}, \Phi)) = \{ \Phi \}$.

In the principal case, $\mathbf{ACC}_{\text{MKNF}}(\mathcal{K})$ contains the belief set $S = \{ \phi \in \Phi \mid \mathcal{M} \models \phi \}$ for some MKNF model $\mathcal{M}$ of $\mathcal{K}$. In that case, $\mathcal{M}$ is the MKNF model of $\mathcal{K}^{\mathcal{M}}$ according to Proposition 44. It is not difficult to verify that $\mathcal{K}^{\mathcal{M}}$ coincides with $\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)$ and that $\mathbf{ACC}_{\text{MKNF}}(\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)) = \{ S \}$.

For the other direction, suppose that $\mathbf{ACC}_{\text{MKNF}}(\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)) = \{ S \}$ for some belief set S. If $S = \Phi$, then $\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)$ coincides with $\mathcal{K}'$ obtained from $\mathcal{K}$ by removing all rules with default negation. This means that $\mathbf{ACC}_{\text{MKNF}}(\mathcal{K}) = \{ S \}$ by definition. In the principal case, $S = \{ \phi \in \Phi \mid \mathcal{M} \models \phi \}$ where $\mathcal{M}$ is the MKNF model of $\mathit{red}_{\text{MKNF}}(\mathcal{K}, S)$ which coincides with $\mathcal{K}^{\mathcal{M}}$. Thus, by Proposition 44, $\mathcal{M}$ is an MKNF model of $\mathcal{K}$ and S belongs to $\mathbf{ACC}_{\text{MKNF}}(\mathcal{K})$ by definition.

It remains to prove that for every MKNF context $C = (L_{\text{MKNF}}, \mathcal{K}, \mathit{br})$, it holds for all $H \subseteq \{ H(\sigma) \mid \sigma \in \mathit{br} \}$ and all belief sets $S \in \mathbf{BS}_{\text{MKNF}}$ that $\mathit{red}_{\text{MKNF}}(\mathcal{K} \cup H, S) = \mathit{red}_{\text{MKNF}}(\mathcal{K}, S) \cup H$. This follows from the fact that all heads of bridge rules from br are objective literals or ontology axioms which are unmodified by the reduction function. □

A.4 Reducing an MKNF Context

In this section we use the multi-context consequence operator that was briefly introduced in [6]. Note that since we assume that the set of belief sets $\mathbf{BS}_i$ is a complete lattice w.r.t. set inclusion, $\mathbf{BS}_1 \times \cdots \times \mathbf{BS}_n$ is also a complete lattice w.r.t. the pointwise ordering $\preceq$. The multi-context consequence operator T_M is a monotonic mapping on this complete lattice, defined as follows:

Definition 45. Let M be a definite MCS. The *multi-context consequence operator* T_M is for any belief state $S = \langle S_1, \dots, S_n \rangle$ of M defined as $T_M(S) = \langle S'_1, \dots, S'_n \rangle$ where

$$\{ S'_i \} = \mathbf{ACC}_i(kb_i \cup \{ H(\sigma) \mid \sigma \in br_i \wedge S \models B(\sigma) \}) \enspace .$$

It is not difficult to verify that T_M is monotonic. Furthermore, the grounded equilibrium of M is the least fixed point of T_M. It also follows from the properties of monotonic mappings on complete lattices that whenever $T_M(S) \preceq S$, the grounded equilibrium S^* of M is such that $S^* \preceq S$.

Proposition 46. *Let M be a definite MCS with the grounded equilibrium S^* and S a belief state of M. If $T_M(S) \preceq S$, then $S^* \preceq S$.*

Proposition 47. *Let $M = \langle C_1, \dots, C_n \rangle$ be a definite MCS such that for some j with $1 \leq j \leq n$, $C_j = (L_{\text{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ is a ground MKNF context, and*

$$M' = \langle C_1, \dots, C_{j-1}, C'_j, C_{j+1}, \dots, C_n \rangle \enspace ,$$

where $C'_j = (L_{\text{MKNF}}, \mathcal{O}, br_j \cup \alpha_j(\mathcal{P}))$. The grounded equilibria of M and M' coincide.

Proof. First let $S = \langle S_1, \dots, S_n \rangle$ be the grounded equilibrium of M. We will show that S is an equilibrium of M'. From the assumption that S is the grounded equilibrium of M we can conclude that, for all i with $1 \leq i \leq n$ and $i \neq j$,

$$\{ S_i \} = \mathbf{ACC}_i(kb_i \cup \{ H(\sigma) \mid \sigma \in br_i \wedge S \models B(\sigma) \}) \enspace . \tag{1}$$

It remains to verify that

$$\{ S_j \} = \mathbf{ACC}_{\text{MKNF}}(\mathcal{O} \cup \{ H(\sigma) \mid \sigma \in br_j \cup \alpha_j(\mathcal{P}) \wedge S \models B(\sigma) \}) \enspace . \tag{2}$$

We know that

$$\{ S_j \} = \mathbf{ACC}_{\text{MKNF}}(\mathcal{O} \cup \mathcal{P} \cup \{ H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \}) \enspace . \tag{3}$$

Let

$$\begin{aligned} \mathcal{O}' &= \mathcal{O} \cup \{ \phi \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) = \phi \text{ is an ontology axiom} \} \enspace , \\ \mathcal{P}' &= \mathcal{P} \cup \{ l \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) = l \text{ is an objective literal} \} \enspace . \end{aligned}$$

It is not difficult to verify that (3) can be written as

$$\{ S_j \} = \mathbf{ACC}_{\text{MKNF}}(\mathcal{O}' \cup \mathcal{P}') \tag{4}$$

while (2) as

$$\{ S_j \} = \mathbf{ACC}_{\text{MKNF}}(\mathcal{O}' \cup \{ H(\pi) \mid \pi \in \mathcal{P}' \wedge S \models \alpha_j(B(\pi)) \}) \enspace . \tag{5}$$

Our goal is to prove that (4) implies (5). First consider the case when $S_j = \Phi$. Then $\mathcal{O}' \cup \mathcal{P}'$ has no MKNF model and by Lemma 40,

$$\begin{aligned} \mathcal{O}' \cup \{ H(\pi) \mid \pi \in \mathcal{P}' \} &= \mathcal{O}' \cup \{ H(\pi) \mid \pi \in \mathcal{P}' \wedge B(\pi) \subseteq S_j \} \\ &= \mathcal{O}' \cup \{ H(\pi) \mid \pi \in \mathcal{P}' \wedge S \models \alpha_j(B(\pi)) \} \end{aligned}$$

also has no MKNF model. Thus, (5) is satisfied. In the principal case there is an MKNF model $\mathcal{M}$ of $\mathcal{O}' \cup \mathcal{P}'$ such that $S_j = \{\, \phi \in \Phi \mid \mathcal{M} \models \phi \,\}$. By Lemma 41, $\mathcal{M}$ is the MKNF model of

$$\begin{aligned}\mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \wedge \mathcal{M} \models \kappa(B(\pi)) \,\} &= \mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \wedge B(\pi) \subseteq S_j \,\} \\ &= \mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \wedge S \models \alpha_j(B(\pi)) \,\} \,.\end{aligned}$$

Hence, (5) is satisfied.

Now suppose that S is the grounded equilibrium of M'. We will show that S is an equilibrium of M. As before, we know that (1) is satisfied, so it remains to verify that (3) is also satisfied given that (2) holds. Similarly as before we can reformulate (3) and (2) as (5) and (4), respectively. So assuming that (5) holds, we need to prove (4).

First suppose that $S_j = \Phi$. It follows that $\mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \,\}$ has no MKNF model. We continue by contradiction. If $\mathcal{O}' \cup \mathcal{P}'$ has the MKNF model $\mathcal{M}$, then for $S'_j = \{\, \phi \in \Phi \mid \mathcal{M} \models \phi \,\}$ we have

$$\{\, S'_j \,\} = \mathbf{ACC}_{\text{MKNF}}(\mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \wedge S \models \alpha_j(B(\pi)) \,\}) \ .$$

Put $S' = \langle S_1, \dots, S_{j-1}, S'_j, S_{j+1}, \dots, S_n \rangle$. Clearly, $S' \prec S$ because S'_j is consistent, so it is a proper subset of $S_j = \Phi$. Thus, by the monotonicity of T_M we obtain $T_M(S') \preceq T_M(S)$. Furthermore, $T_M(S) = S'$, so we can conclude that $T_M(S') \preceq S'$ and by Proposition 46 we conclude that for the grounded equilibrium S^* of M it holds that $S^* \preceq S'$. It also follows from the above that $S^* \prec S$. Moreover, by the first part of this proof we conclude that S^* is an equilibrium of M', which is in conflict with the minimality of S.

A similar argument applies when $S_j = \{\, \phi \in \Phi \mid \mathcal{M} \models \phi \,\}$ for some MKNF model $\mathcal{M}$ of $\mathcal{O}' \cup \{\, H(\pi) \mid \pi \in \mathcal{P}' \wedge S \models \alpha_j(B(\pi)) \,\}$. It follows from Lemma 41 that $\mathcal{M}$ is an S5 model of $\mathcal{O}' \cup \mathcal{P}'$ and thus by Proposition 39, $\mathcal{O}' \cup \mathcal{P}'$ has an MKNF model $\mathcal{M}' \supseteq \mathcal{M}$. Let $S'_j = \{\, \phi \in \Phi \mid \mathcal{M}' \models \phi \,\}$. It easily follows that $S'_j \subseteq S_j$. Put $S' = \langle S_1, \dots, S_{j-1}, S'_j, S_{j+1}, \dots, S_n \rangle$. Clearly, $S' \preceq S$. Thus by the monotonicity of T_M we obtain $T_M(S') \preceq T_M(S)$. Furthermore, $T_M(S) = S'$, so we can conclude that $T_M(S') \preceq S'$ and by Proposition 46 we conclude that for the grounded equilibrium S^* of M it holds that $S^* \preceq S'$. It also follows from the above that $S^* \preceq S$. Moreover, by the first part of this proof we conclude that S^* is an equilibrium of M', and by the minimality of S we obtain $S \preceq S^*$. Consequently, $S = S^* = S'$.

Finally, since the grounded equilibrium of M is an equilibrium of M' and the grounded equilibrium of M' is an equilibrium of M, it follows by the minimality of grounded equilibria that the grounded equilibria of M and M' coincide. □

Proposition 48. *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible MCS such that for some j with $1 \leq j \leq n$, $C_j = (L_{\text{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ is a ground MKNF context, and*

$$M' = \langle C_1, \dots, C_{j-1}, C'_j, C_{j+1}, \dots, C_n \rangle \ ,$$

where $C'_j = (L_{\text{MKNF}}, \mathcal{O}, br_j \cup \alpha_j(\mathcal{P}))$. The grounded equilibria of M and M' coincide.

Proof. This follows from the definition of α_j, the reduct of bridge rules br_j^S and logic programs $\mathcal{P}^{S_j}$ and the fact that for every belief state S, the pair of definite multi-context systems M^S, $(M')^S$ satisfies the conditions of Proposition 47. □

Theorem 16 (Reduction Into First-Order Context). *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible MCS such that for some j with $1 \le j \le n$, C_j is a ground MKNF context, and*

$$M' = \langle C_1, \dots, C_{j-1}, C_j^{\text{FO}}, C_{j+1}, \dots, C_n \rangle \ .$$

The grounded equilibria of M and M' coincide.

Proof. This follows from Proposition 48 and from [10, Theorem 5.9.4] together with the fact that we work under the UNA, so equality is interpreted the same way for first-order and MKNF theories. □

A.5 Reduction to DL + Database

The proof proceeds in three steps. At first, we show that, starting from a first-order context which resulted from an MKNF context, we can obtain two corresponding contexts, one FO context and one database context.

Definition 49. Let $C_j = (L_{\text{FO}}, \{\, \kappa(\mathcal{O}) \,\}, br_j)$ be the first-order context corresponding to a ground MKNF context. The *two-context MCS corresponding to C_j*, $\langle C'_j, C'_k \rangle$, is defined as follows:

- $C'_j = (L_{\text{FO}}, \{\kappa(\mathcal{O})\}, br'_j)$, where

$$br'_j = \{\, \sigma \mid \sigma \in \beta_j^k(br_j) \wedge H(\sigma) \text{ resulted from applying } \kappa \,\} \ ;$$

- $C'_k = (L_{\text{DB}}, \emptyset, br'_k)$, where

$$br'_k = \{\, \sigma \mid \sigma \in \beta_j^k(br_j) \wedge H(\sigma) \text{ is a non-DL-literal} \,\} \ .$$

Proposition 50. *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible multi-context system such that, for some j with $1 \le j \le n$, C_j is a first-order context corresponding to a ground MKNF context, and*

$$M' = \langle C'_1, \dots, C'_{j-1}, C'_j, C'_{j+1}, \dots, C'_n, C'_{n+1} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \le i \le n$ and $i \ne j$, $C'_i = (L_i, kb_i, \beta_j^{n+1}(br_i))$. The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \dots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \dots, S'_{j-1}, S_j^{\text{DL}}, S'_{j+1}, \dots, S'_n, S_{n+1}^{\text{DB}} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \le i \le n$ and $i \ne j$, $S_i = S'_i$, and $S_j = (S_j^{\text{DL}} \cup S_{n+1}^{\text{DB}})^*$.

Proof. First, let $S = \langle S_1, \dots, S_n \rangle$ be a grounded equilibrium of M. We show that $S' = \langle S'_1, \dots, S'_{n+1} \rangle$ is an equilibrium of M'. Since S is a grounded equilibrium of M we know that, for all i with $1 \le i \le n$,

$$\{\, S_i \,\} = \mathbf{ACC}_i(kb_i \cup \{\, H(\sigma) \mid \sigma \in br_i \wedge S \models B(\sigma) \,\})$$

Consider the first-order context C_j corresponding to a finite ground MKNF context:

$$\{\, S_j \,\} = \mathbf{ACC}_j(\kappa(\mathcal{O}) \cup \{\, H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \,\})$$

where rules in br_j can be divided into rules whose head resulted from applying κ and rules with a non-DL-literal in the head. Thus the following is equivalent:

$$\begin{aligned}
\{\, S_j \,\} = \mathbf{ACC}_j(&\kappa(\mathcal{O}) \cup \\
&\{\, H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) \text{ resulted from applying } \kappa \,\} \cup \\
&\{\, H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) \text{ is a non-DL-literal} \,\})
\end{aligned}$$

Since non-DL-atoms do not appear in $\mathcal{O}$, and UNA prevents any further derivations due to equalities in $\mathcal{O}$, we can rewrite this (for some integer k) as follows: $S_j = (S'_j \cup S'_k)^*$ where

$$\begin{aligned}\{\, S'_j \,\} &= \mathbf{ACC}_j(\kappa(\mathcal{O})\cup \\ &\qquad \{\, H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) \text{ resulted from applying } \kappa \,\}) \\ \{\, S'_k \,\} &= \mathbf{ACC}_j(\{\, H(\sigma) \mid \sigma \in br_j \wedge S \models B(\sigma) \wedge H(\sigma) \text{ is a non-DL-literal} \,\}))^*\end{aligned}$$

Note that the deductive closure is necessary to capture first-order sentences in S_j that contain elements from both its parts.

Recall, that all bridge literals referring to C_j are objective literals or ontology axioms to which κ has been applied, i.e. one of the two contexts suffices to verify whether such a bridge literal holds. Moreover, note that S'_k alone is just a set of facts and does not require a first-order context. Thus, we can introduce the two-context MCS corresponding to C_j, $\langle C'_j, C'_k \rangle$ to which $\langle S'_j, S'_k \rangle$ corresponds, and, for $k = n + 1$, obtain M' by applying β^k_j to all the bridge rules in each context in M, which simply redirects bridge literals that refer to non-DL-literals to C'_{n+1}.

We obtain immediately that, for all i with $1 \leq i \neq j \leq n$, $S_i = S'_i$, and also $S_j = (S'_j \cup S'_{n+1})^*$, and, subsequently, that, for all $1 \leq i' \leq n + 1$,

$$\{\, S_{i'} \,\} = \mathbf{ACC}_{i'}(kb_{i'} \cup \{\, H(\sigma) \mid \sigma \in br_{i'} \wedge S \models B(\sigma) \,\})$$

Consequently, S' is an equilibrium of M'.

Now, let $S' = \langle S'_1, \ldots, S'_{n+1} \rangle$ be a grounded equilibrium of M'. We can show that $S = \langle S_1, \ldots, S_n \rangle$ is an equilibrium of M following the (inverted) constructive argument of the previous part.

Finally, since a grounded equilibrium $\langle S_1, \ldots, S_n \rangle$ of M yields an equilibrium $\langle S'_1, \ldots, S'_{n+1} \rangle$ of M', and a grounded equilibrium $\langle S'_1, \ldots, S'_{n+1} \rangle$ of M' yields an equilibrium $\langle S_1, \ldots, S_n \rangle$ of M, such that, for all i with $1 \leq i \neq j \leq n$, $S_i = S'_i$, and $S_j = (S'_j \cup S'_{n+1})^*$, we conclude by the minimality of grounded equilibria that the grounded equilibria of M and M' are equivalent. □

In the next step, the transformation is lifted to work for an MKNF context directly.

Definition 51. Let $C_j = (L_{\text{MKNF}}, \mathcal{O} \cup \mathcal{P}, br_j)$ be a ground MKNF context. The *two-context FO MCS corresponding to* C_j, $\langle C^{\text{FOL}}_j, C^{\text{DB}}_k \rangle$, is defined as follows:

- $C^{\text{FOL}}_j = (L_{\text{FO}}, \{\kappa(\mathcal{O})\}, br^{\text{FOL}}_j)$, where
$$br^{\text{FOL}}_j = \{\, \sigma \mid \sigma \in \beta^k_j(br_j \cup \alpha_j(\mathcal{P})) \wedge H(\sigma) \text{ resulted from applying } \kappa \,\} \quad ;$$
- $C^{\text{DB}}_k = (L_{\text{DB}}, \emptyset, br_k)$, where
$$br_k = \{\, \sigma \mid \sigma \in \beta^k_j(br_j \cup \alpha_j(\mathcal{P})) \wedge H(\sigma) \text{ is a non-DL-literal } \,\} \quad .$$

Note that C^{FOL}_j is different from C^{FO}_j in Theorem 16.

Proposition 52. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible multi-context system such that, for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and set*

$$M' = \langle C'_1, \ldots, C'_{j-1}, C^{\text{FOL}}_j, C'_{j+1}, \ldots, C'_n, C^{\text{DB}}_{n+1} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ and $i \neq j$, $C'_i = (L_i, kb_i, \beta^{n+1}_j(br_i))$.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S^{\text{DL}}_j, S'_{j+1}, \ldots, S'_n, S^{\text{DB}}_{n+1} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq j$, $S_i = S'_i$, and $S_j = (S^{\text{DL}}_j \cup S^{\text{DB}}_{n+1})^*$.

Proof. This follows directly from Proposition 50 since, despite the differing original context C_j (ground MKNF context in Proposition 52 vs. a first-order context corresponding to a ground MKNF context in Proposition 50), both two-context MCS $\langle C'_j, C'_k \rangle$ and $\langle C^{\text{FOL}}_j, C^{\text{DB}}_k \rangle$ are identical. □

Theorem 20. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible MCS such that, for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and*

$$M' = \langle C'_1, \ldots, C'_{j-1}, C^{\text{DL}}_j, C'_{j+1}, \ldots, C'_n, C^{\text{DB}}_{n+1} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ and $i \neq j$, $C'_i = (L_i, kb_i, \beta^{n+1}_j(br_i))$.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S^{\text{DL}}_j, S'_{j+1}, \ldots, S'_n, S^{\text{DB}}_{n+1} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq j$, $S_i = S'_i$, and $S_j = (S^{\text{DL}}_j \cup S^{\text{DB}}_{n+1})^*$.

Proof. This follows directly from Proposition 52. The only differences are that C^{FOL}_j uses $\{\kappa(\mathcal{O})\}$ as kb instead of $\mathcal{O}$, since C^{FOL}_j uses L_{FO} instead of L_{DL}, and that some bridge rule heads are results of applying κ in the former case instead of ontology axioms, which can be remedied by a simple inverse function κ^-. □

A.6 Reduction to DL + ASP

In the following, we proceed as in the previous subsection. We first define an intermediate translation from DL-DB MCS to DL-ASP MCS. For that purpose, we define the DL-ASP MCS corresponding to a DL-DB MCS.

Definition 53. Let $\langle C^{\text{DL}}_j, C^{\text{DB}}_k \rangle = \langle (L_{\text{DL}}, \mathcal{O}, br^{\text{DL}}_j), (L_{\text{DB}}, \emptyset, br^{\text{DB}}_k) \rangle$ be a DL-DB MCS corresponding to a ground MKNF context. The DL-ASP MCS corresponding to $\langle C^{\text{DL}}_j, C^{\text{DB}}_k \rangle$, $\langle C^{\text{DL}m}_j, C^{\text{ASP}}_k \rangle$, is defined as follows:

- $C^{\text{DL}m}_j = (L_{\text{DL}}, \mathcal{O}, br^{\text{DL}m}_j)$, where

$$br^{\text{DL}m}_j = \{\sigma \mid \sigma \in (br^{\text{DL}}_j)^m\} \cup \{H(\sigma) \leftarrow (k : \gamma_j(H(\sigma))). \mid \sigma \in (br^{\text{DL}}_j)^n\} \ ;$$

- $C^{\text{ASP}}_k = (L_{\text{ASP}}, kb_k, br^{\text{ASP}}_k)$, where

$$\begin{aligned} kb_k = \{\, & H(\sigma) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in (br^{\text{DB}}_k)^n \,\} \\ & \cup \{\, \gamma_j(H(\sigma)) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in (br^{\text{DL}}_j)^n \,\} \ , \end{aligned}$$

$$\begin{aligned} br^{\text{ASP}}_k = \{\, \sigma \mid \sigma \in (br^{\text{DB}}_k)^m \,\} \cup \{\, \gamma_i(p) \leftarrow (i : p). \mid \sigma \in ((br^{\text{DL}}_j)^n \cup (br^{\text{DB}}_k)^n) \\ \wedge k \neq i \wedge (i : p) \in B(\sigma)^+ \cup B(\sigma)^- \,\} \ . \end{aligned}$$

Proposition 54. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible multi-context system such that, for some j and k with $1 \leq j \neq k \leq n$, $\langle C^{\text{DL}}_j, C^{\text{DB}}_k \rangle$ is a DL-DB MCS corresponding to a ground MKNF context, and $M' = \langle C'_1, \ldots, C'_n \rangle$ where, for all i with $1 \leq i \leq n$ and $j \neq i \neq k$, $C'_i = C_i$, $C'_j = C^{\text{DL}m}_j$ and $C'_k = C^{\text{ASP}}_k$.*

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_n \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq k$, $S_i = S'_i$, and $S_k = S'_k \setminus S^{\text{AUX}}$ where $S^{\text{AUX}} \subseteq S'_k$ is the set of all new ground atoms introduced by γ.

Proof. First, let $S = \langle S_1, \ldots, S_n \rangle$ be a grounded equilibrium of M. We show that $S' = \langle S'_1, \ldots, S'_n \rangle$ is an equilibrium of M'. Since S is a grounded equilibrium of M we know that, for all i with $1 \leq i \leq n$,

$$\{ S_i \} = \mathbf{ACC}_i(kb_i \cup \{ H(\sigma) \mid \sigma \in br_i \wedge S \models B(\sigma) \})$$

Consider the DL-DB MCS $\langle C_j^{\text{DL}}, C_k^{\text{DB}} \rangle$ corresponding to a ground MKNF context:

$$\begin{aligned} \{ S_j \} &= \mathbf{ACC}_j(\mathcal{O} \cup \{ H(\sigma) \mid \sigma \in br_j^{\text{DL}} \wedge S \models B(\sigma) \}) \\ \{ S_k \} &= \mathbf{ACC}_k(\{ H(\sigma) \mid \sigma \in br_k^{\text{DB}} \wedge S \models B(\sigma) \}) \end{aligned}$$

We can distinguish monotonic and non-monotonic bridge rules as follows without any effect on the belief sets.

$$\begin{aligned} \{ S_j \} = \mathbf{ACC}_j(\mathcal{O} \cup \{ H(\sigma) \mid \sigma \in (br_j^{\text{DL}})^m \wedge S \models B(\sigma) \} \cup \\ \{ H(\sigma) \mid \sigma \in (br_j^{\text{DL}})^n \wedge S \models B(\sigma) \}) \\ \{ S_k \} = \mathbf{ACC}_k(\{ H(\sigma) \mid \sigma \in (br_k^{\text{DB}})^m \wedge S \models B(\sigma) \} \cup \\ \{ H(\sigma) \mid \sigma \in (br_k^{\text{DB}})^n \wedge S \models B(\sigma) \}) \end{aligned}$$

We can also switch the part of S_j resulting from non-monotonic bridge rules to S_k using ground atoms $\gamma_j(H(\sigma))$ as links in the additional monotonic bridge rules from C_k to C_j.

$$\begin{aligned} \{ S_j \} = \mathbf{ACC}_j(\mathcal{O} \cup \{ H(\sigma) \mid \sigma \in (br_j^{\text{DL}})^m \wedge S \models B(\sigma) \} \cup \\ \{ H(\sigma) \mid \gamma_j(H(\sigma)) \in S_k \wedge \sigma \in (br_j^{\text{DL}})^n \}) \\ \{ S''_k \} = \mathbf{ACC}_k(\{ H(\sigma) \mid \sigma \in (br_k^{\text{DB}})^m \wedge S \models B(\sigma) \} \cup \\ \{ H(\sigma) \mid \sigma \in (br_k^{\text{DB}})^n \wedge S \models B(\sigma) \} \cup \\ \{ \gamma_j(H(\sigma)) \mid \sigma \in (br_j^{\text{DL}})^n \wedge S \models B(\sigma) \}) \end{aligned}$$

This step does obviously not change S_j but potentially adds new ground atoms to S''_k, so $S_k \subseteq S''_k$.

Now, we join the two belief sets based on the of non-monotonic bridge rules and turn them into ASP rules using γ. We only have to ensure that the necessary information for the bridge literals from other contexts is correctly imported, where again γ ensures that only new ground atoms are introduced.

$$\begin{aligned} \{ S_j \} = \mathbf{ACC}_j(\mathcal{O} \cup \{ H(\sigma) \mid \sigma \in (br_j^{\text{DL}})^m \wedge S \models B(\sigma) \} \cup \\ \{ H(\sigma) \mid \gamma_j(H(\sigma)) \in S_k \wedge \sigma \in (br_j^{\text{DL}})^n \}) \\ \{ S''_k \} = \mathbf{ACC}_k(\{ H(\sigma) \mid \sigma \in (br_k^{\text{DB}})^m \wedge S \models B(\sigma) \} \cup \\ \{H(\sigma) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in (br_k^{\text{DB}})^n\} \cup \\ \{\gamma_j(H(\sigma)) \leftarrow \gamma_k(B(\sigma)). \mid \sigma \in (br_j^{\text{DL}})^n\} \cup \\ \{\gamma_i(p) \mid p \in S_i \wedge \sigma \in ((br_j^{\text{DL}})^n \cup (br_k^{\text{DB}})^n) \wedge k \neq i \\ \wedge (i : p) \in B(\sigma)^+ \cup B(\sigma)^- \}) \end{aligned}$$

We derive from this that $S_j = S'_j$ and $S''_k = S'_k$. Indeed, $S_k = S'_k \setminus S^{\text{AUX}}$ holds.

Now, since all elements added to S'_k are new ground atoms, we can be sure that their addition to S_k does not yield any additional information in any other context (apart from the intended transfer for $\gamma_j(H(\sigma))$ to S_j). We conclude that S' is an equilibrium of M'.

Assuming that S' is a grounded equilibrium of M', we can also show that S is an equilibrium of M following the inverse of the just presented construction. By minimality, we obtain that both obtained equilibria are indeed grounded equilibria, which finishes the proof. □

Theorem 25. *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible multi-context system such that, for some j with $1 \leq j \leq n$, C_j is a ground MKNF context, and*

$$M' = \langle C'_1, \dots, C'_{j-1}, C_j^{\text{DL}_m}, C'_{j+1}, \dots, C'_n, C_{n+1}^{\text{ASP}} \rangle$$

where, for all $C_i = (L_i, kb_i, br_i)$ with $1 \leq i \leq n$ and $i \neq j$, $C'_i = (L_i, kb_i, \beta_j^{n+1}(br_i))$.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \dots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \dots, S'_{j-1}, S_j^{\text{DL}_m}, S'_{j+1}, \dots, S'_n, S_{n+1}^{\text{ASP}} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \neq j \leq n$, $S_i = S'_i$, and $S_j = (S_j^{\text{DL}_m} \cup (S_{n+1}^{\text{ASP}} \setminus S^{\text{AUX}}))^*$ where $S^{\text{AUX}} \subseteq S_{n+1}^{\text{ASP}}$ is the set of all new ground atoms introduced by γ.

Proof. Consider Defs. 19 and 53 for the step-wise correspondences from MKNF context via DL-DB MCS to DL-ASP MCS in comparison with Def. 24 for the direct correspondence from MKNF context to DL-ASP MCS. Consider in particular the two resulting DL-ASP MCS in Defs. 53 and 24. They are indeed identical in every aspect but one: in Def. 53, $br_j^{\text{DL}_m}$ contains also the monotonic rules from $\mathcal{P}$ of the MKNF KB $\mathcal{K}$, while in Def. 24 these rules are added to the kb of the ASP context. The equivalence of the grounded equilibria of M and M' thus follows from a similar argument as in Prop. 54, Theorem 20, and Prop. 54 itself. □

Theorem 28. *Let $M = \langle C_1, \dots, C_n \rangle$ be a reducible MCS with $C_i = (L_i, kb_i, br_i)$ for all i with $1 \leq i \leq n$ and C_j a ground ASP context for some j with $1 \leq j \leq n$, and $M' = \langle C'_1, \dots, C'_n \rangle$ with*

- *$C'_i = (L_i, kb_i, br'_i)$ for all i with $i \neq j$, where*
  $$br'_i = br_i^m \cup \{\, H(\sigma) \leftarrow (j : \gamma_i(H(\sigma))). \mid \sigma \in br_i^n \,\} \quad ;$$
- *$C'_j = (L_j, kb'_j, br'_j)$, where $kb'_j = kb_j \cup \bigcup_i \{\gamma_i(H(\sigma)) \leftarrow \gamma_j(B(\sigma)). \mid \sigma \in br_i^n\}$, and*
  $$br'_j = br_j^m \cup \bigcup_i \{\, \gamma_k(p) \leftarrow (k : p). \mid \sigma \in br_i^n \wedge j \neq k \wedge (k : p) \in B(\sigma)^+ \cup B(\sigma)^- \,\} \quad .$$

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \dots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \dots, S'_n \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i \leq n$ and $i \neq j$, $S_i = S'_i$, and $S_j = S'_j \setminus S^{\text{AUX}}$ where $S^{\text{AUX}} \subseteq S'_j$ is the set of all new ground atoms introduced by γ.

Proof. This theorem is indeed a generalisation of Proposition 54. Indeed, the DL-ASP MCS defined in Definition 53 contains $C_j^{\text{DL}_m}$ which exactly corresponds to all C'_i whose non-monotonic bridge rules are transferred to the ASP context. Moreover, C_j is the ASP context C_k^{ASP}, only that now kb contains the union of all non-monotonic (transformed) bridge rules, and the bridge rules itself are also a union of imports from all different contexts. Note that since new ground atoms are disjoint with already elements of the belief sets, the transfer of additional beliefs into the ASP context does not affect the equilibria. □

A.7 Reduction to MKNF Context

Theorem 31. *Let $M = \langle C_1, \ldots, C_n \rangle$ be a reducible multi-context system such that, for some j and k with $1 \leq j < k \leq n$, C_j and C_k are ground MKNF contexts, and*

$$M' = \langle C'_1, \ldots, C'_{j-1}, C_j^{\text{MKNF}}, C'_{j+1}, \ldots, C'_{n-1} \rangle$$

where, given $C_i = (L_i, kb_i, br_i)$, for all i with $1 \leq i < n$ and $i \neq j$, $C'_i = (L_i, kb_i, \delta_j^k(br_i))$ if $i < k$, and $C'_i = (L_{i+1}, kb_{i+1}, \delta_j^k(br_{i+1}))$ otherwise.

The grounded equilibria of M and M' are equivalent, i.e. $S = \langle S_1, \ldots, S_n \rangle$ is a grounded equilibrium for M iff $S' = \langle S'_1, \ldots, S'_{j-1}, S_j^{\text{MKNF}}, S'_{j+1}, \ldots, S'_{n-1} \rangle$ is a grounded equilibrium for M' where, for all i with $1 \leq i < n$ and $i \neq j$, $S'_i = S_i$ if $i < k$ and $S'_i = S_{i+1}$ otherwise, and $S_j^{\text{MKNF}} = (S_j \cup S_k)^*$.

Proof. First, let $S = \langle S_1, \ldots, S_n \rangle$ be a grounded equilibrium of M. We show that $S' = \langle S'_1, \ldots, S'_n \rangle$ is an equilibrium of M'.

Let C_j and C_k be ground MKNF contexts. The corresponding belief sets are:

$$\begin{aligned}\{\, S_j \,\} &= \mathbf{ACC}_j(\mathcal{O}^1 \cup \mathcal{P}^1 \cup \{\, H(\sigma) \mid \sigma \in br_j^1 \wedge S \models B(\sigma) \,\})\\ \{\, S_k \,\} &= \mathbf{ACC}_k(\mathcal{O}^2 \cup \mathcal{P}^2 \cup \{\, H(\sigma) \mid \sigma \in br_k^2 \wedge S \models B(\sigma) \,\})\end{aligned}$$

Since, by assumption, all predicates appearing in the two contexts are disjoint, we obtain that

$$\begin{aligned}\{\, (S_j \cup S_k)^* \,\} = \mathbf{ACC}_{\text{MKNF}}(&\mathcal{O}^1 \cup \mathcal{P}^1 \cup \mathcal{O}^2 \cup \mathcal{P}^2\\ &\cup \{\, H(\sigma) \mid \sigma \in (br_j^1 \cup br_k^2) \wedge S \models B(\sigma) \,\}) \ .\end{aligned}$$

This exactly corresponds to S_j^{MKNF} apart from the reordering δ. Now, since the other belief sets are not changed at all (apart from the reordering), we derive that S' is an equilibrium of M'.

Assuming that S' is a grounded equilibrium of M' , we can show S is an equilibrium of M following the inverse of the constructive argument just presented.

By minimality, we obtain that both obtained equilibria are indeed grounded equilibria, which finishes the proof. □