Knowledge-Based Programs with Defaults in a Modal Situation Calculus

Jens Claßen[1] and **Malte Neuss**[2]

Abstract. We consider the realistic case of a GOLOG agent that only possesses incomplete knowledge about the state of its environment and has to resort to sensing in order to gather additional information at runtime, and where the agent is controlled by a knowledge-based program in which test conditions explicitly refer to the agent's knowledge (or lack thereof). In this paper, we propose a formalization of knowledge-based agents that extends earlier proposals by a form of non-monotonic reasoning that includes Reiter-style defaults. We present a reasoning mechanism that enables us to reduce projection queries about future states of the agent's knowledge (including nesting of epistemic modalities) to classical Default Logic, and provide a corresponding Representation Theorem. We thus obtain the theoretical foundation for an implementation where reasoning subtasks can be handed to an embedded off-the-shelf reasoner for Default Logic, and that supports a (in some respects) more expressive epistemic action language than previous solutions.

1 INTRODUCTION

The GOLOG [19, 4] family of action languages and the underlying Situation Calculus action formalism [25, 32] are a popular means for the high-level control of autonomous agents. Reiter [33] proposed an extension for realistic scenarios where the agent possesses only incomplete information about its surroundings and has to resort to runtime sensing to fulfill its task. In what is called a *knowledge-based program*, test conditions may then explicitly refer to the agent's epistemic state, thus enabling it to reason about its own knowledge (or lack thereof). Based on Scherl and Levesque's [36] epistemic extension of the Situation Calculus, he furthermore provided a regression-based reasoning procedure where deciding subjective queries about future situations is reduced to standard theorem proving.

Nevertheless, Reiter's approach came with some limitations. On the one hand, the notion of *only knowing* [17] – the fact that the agent's knowledge base represents *all and only* what the agent knows – is only formalized in a meta-theoretic manner, which makes theoretical considerations difficult. More importantly, despite the ability to explicitly refer to the agent's knowledge, he considers a very limited set of queries that disallows referring to future situations, nesting of modal knowledge operators (needed for introspection) and quantifying-in (necessary to distinguish "knowing that" from "knowing what"). Consider the well-known Wumpus domain [34], where the agent acts in a grid world whose cells (called *rooms*) may be occupied by pits (e.g. squares $\langle 3,1\rangle$ and $\langle 4,4\rangle$ in Figure 1) or the

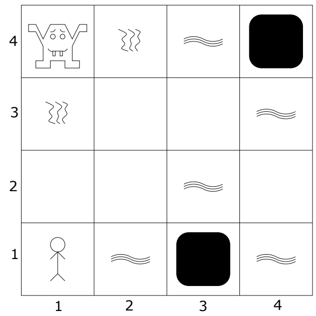

Figure 1. Example Wumpus world

Wumpus (square $\langle 1,4\rangle$), each of which are lethal (entering such a room causes the agent to die instantly). Initially, the agent is unaware about the locations of pits and Wumpus, but in adjacent squares it can sense a breeze (in the case of pits) or a stench (in the case of the Wumpus) and use that information to infer which grid cells are safe.

The basic reasoning task during the execution of a knowledge-based program that controls such an agent is to solve projection queries, i.e. to decide whether some formula about the agent's knowledge after the execution of certain actions holds or not. For example,

$$[move(north)]\mathbf{K}([sensebreeze]\exists r\mathbf{K}(Safe(r))) \tag{1}$$

expresses that after moving north, it is known that when a $sensebreeze$ action is performed, the agent will come to know a room that is safe. Claßen and Lakemeyer [3] proposed a new formalization of knowledge-based programs based on the modal Situation Calculus variant $\mathcal{ES}$ [11, 10] that includes a modal operator $\mathbf{O}$ for only knowing and that resorts to Levesque and Lakemeyer's [18] Representation Theorem in order to be able to evaluate conditions such as the above.

So far, this approach requires the agent's knowledge base to only contain objective statements about the world such as the fact that a room r is only safe if it does not contain a pit or the Wumpus:

$$\neg Pit(r) \wedge \neg Wumpus(r) \supset Safe(r) \tag{2}$$

However, often we may want to exploit more of the expressiveness of $\mathcal{ES}$ and add non-objective formulas that represent defaults, and hence enable non-monotonic reasoning. In the example, we could say that any room that is potentially unsafe should be considered dangerous (and thus avoided):

$$\mathbf{K}(Room(r)) \wedge \mathbf{M}(\neg Safe(r)) \supset Dangerous(r) \tag{3}$$

Whereas $\mathbf{K}(\alpha)$ is to be read as "α is known", $\mathbf{M}(\alpha)$ means "α is consistent with what is known". $Dangerous(r)$ then is a default conclusion that may later be withdrawn in face of new information obtained

[1] Knowledge-Based Systems Group, RWTH Aachen University, Germany, email: classen@kbsg.rwth-aachen.de
[2] Knowledge-Based Systems Group, RWTH Aachen University, Germany, email: malte.neuss@rwth-aachen.de

through sensing: If the agent neither senses a breeze nor a stench near r, it concludes that that room neither contains a pit nor the Wumpus, and hence does not regard it as dangerous anymore.

The main contribution of this paper is a new variant of Levesque and Lakemeyer's Representation Theorem for knowledge bases that may contain such non-objective default rules. In the same way that their method for objective knowledge bases reduces reasoning about action and knowledge to standard first-order theorem proving, ours will constitute a similar reduction to a standard non-monotonic reasoning method. In particular, we are interested in compatibility to Reiter's Default Logic [30], which will allow us to resort to existing off-the-shelf reasoners for this formalism, including solvers for Answer Set Programming (ASP) [6] due to the well-known relationship to normal logic programs under the stable model semantics [7].

For this purpose, we combine $\mathcal{ES}$ with $\mathcal{O}_3\mathcal{L}$ [12, 13], a logic that establishes an exact correspondence between three variants of the only-knowing operator $\mathbf{O}$ (along with the meaning of $\mathbf{M}$) and different non-monotonic logics. In particular, Levesque's original definition of only-knowing [18] corresponds to Moore's Autoepistemic Logic [26], in which case $\mathbf{M}(\alpha)$ is interpreted as $\neg\mathbf{K}(\neg\alpha)$. Moreover, varying the semantics of $\mathbf{O}$ enables to capture Reiter's Default Logic, where the duality between $\mathbf{M}$ and $\mathbf{K}$ is given up, intuitively because "ungrounded" extensions are treated differently.[3]

The remainder of this paper is organized as follows. In Section 2, we introduce our new logic called $\mathcal{ESD}$ that can be viewed as amalgamation of Lakemeyer and Levesque's $\mathcal{ES}$ (which supports action, sensing, but no Reiter-style defaults) with the default part of their logic $\mathcal{O}_3\mathcal{L}$ (which does not include actions and sensing). We then proceed to show that the new logic is a unifying formalism that serves as a coherent foundation for the definition of knowledge-based agents. For this purpose, in Section 3 we extend the standard Situation Calculus and $\mathcal{ES}$ definitions of basic action theories (used for axiomatizing dynamic domains) to the $\mathcal{ESD}$ case, and present a corresponding regression operator (used for reducing reasoning about actions to reasoning about the initial situation). Section 4 then contains our main contribution in form of a new variant of Levesque and Lakemeyer's Representation Theorem for eliminating knowledge operators, now also accounting for defaults in the agent's knowledge base. Section 5 then discusses how to combine all above mentioned results in order to implement a knowledge-based GOLOG agent capable of default reasoning. Finally, we review related work and conclude.

2 THE LOGIC $\mathcal{ESD}$

In this section we define a dynamic logic of only-knowing and default reasoning. It can be viewed as an amalgamation of the Default Logic part of Lakemeyer and Levesque's static logic $\mathcal{O}_3\mathcal{L}$ [12, 13] with their modal epistemic Situation Calculus variant $\mathcal{ES}$ [11].

2.1 Syntax

The alphabet of $\mathcal{ESD}$ consists of the usual logical connectives and quantifiers, punctuation, parentheses, a countably infinite supply of first-order variables, equality, rigid predicates of any arity, fluent predicates of any arity (including the special predicates $Poss$ and SF for action preconditions and sensing, respectively), a countably infinite set of standard names (which are syntactically treated as constants), the modal operators $[\cdot]$ and $\Box$ as well as the epistemic modal operators $\mathbf{K}$, $\mathbf{M}$, $\mathbf{O}_R$ and $\mathbf{O}_M$. The *terms* are the variables and standard names. Formulas are defined inductively as follows:

- if $t_1, \ldots, t_k$ are terms and P is a predicate of arity k, then $P(t_1, \ldots, t_k)$ is an (atomic) formula;
- if t_1 and t_2 are terms, then $(t_1 = t_2)$ is a formula;
- if α and β are formulas, x is a variable, and t a term, then $\neg\alpha$, $(\alpha \wedge \beta)$, $\forall x\alpha$, $\mathbf{K}(\alpha)$, $\mathbf{M}(\alpha)$, $\mathbf{O}_M(\alpha)$, $\mathbf{O}_R(\alpha)$, $[t]\alpha$ and $\Box\alpha$ are also formulas.

Intuitively, $\mathbf{K}(\alpha)$ is read as "α is known by the agent", while $\mathbf{M}(\alpha)$ means "α is consistent with what the agent knows". Both $\mathbf{O}_M(\alpha)$ and $\mathbf{O}_R(\alpha)$ are to be read as "α is all that is known", with the difference that with $\mathbf{O}_M$, defaults will be evaluated according to Moore's Autoepistemic Logic, whereas $\mathbf{O}_R$ corresponds to Reiter's Default Logic. Finally, $[t]\alpha$ means "α holds after executing action t" and $\Box\alpha$ stands for "α holds after any number of actions". Also note that for simplicity we do not distinguish sorts, but allow any term to be used as an action.

We treat $(\alpha \vee \beta)$, $(\alpha \supset \beta)$, $\exists x\alpha$, truth $\top$, and falsity $\bot$ as the usual abbreviations. For a finite sequence $z = \langle n_1, \ldots, n_k \rangle$ of actions, we let $[z]\alpha$ stand for $[n_1] \cdots [n_k]\alpha$. The notion of free and bound variables is defined in the usual way, and α^x_t means α with all free occurrences of x replaced by t. A formula without free variables is called a *sentence*, and an atomic formula $P(n_1, \ldots, n_k)$ where all n_i are standard names is called *primitive sentence*. Formulas without any occurrence of epistemic modal operators are called *objective*, those where all predicates appear within the scope of an epistemic modal operator *subjective*, those without $\mathbf{O}_R$ and $\mathbf{O}_M$ *basic*, those without $[\cdot]$ and $\Box$ *static*, and those without $\Box$ *bounded*. A *fluent* formula is one that is objective, static and neither mentions $Poss$ nor SF.

2.2 Semantics

The semantics is as follows. Let $\mathcal{N}$ denote the set of all standard names and $\mathcal{Z}$ the set of all finite sequences z of standard names, including the empty sequence $\langle\rangle$. A *world* w is given by a mapping from the primitive sentences and $\mathcal{Z}$ to truth values $\{0, 1\}$, respecting *rigidity*, i.e. if R is a rigid predicate, then for all $z, z' \in \mathcal{Z}$, $w[R(\vec{n}), z] = w[R(\vec{n}), z']$. Let $\mathcal{W}$ be the set of all worlds. An *epistemic state* e is given by a set of worlds, i.e. a subset of $\mathcal{W}$. For two worlds w and w' and a sequence $z \in \mathcal{Z}$, $w \simeq_z w'$ (read: w and w' agree on the sensing for z) is inductively defined as follows:

- $w \simeq_{\langle\rangle} w'$ for every w and w';
- $w \simeq_{z \cdot n} w'$
  iff $w \simeq_z w'$ and $w[SF(n), z] = w'[SF(n), z]$.

We are now ready to define the truth of sentences. $\mathcal{ESD}$, similar as $\mathcal{O}_3\mathcal{L}$, uses two epistemic states to interpret formulas, one to interpret formulas with $\mathbf{K}$, the other to interpret formulas with $\mathbf{M}$. We need this distinction because the duality $\mathbf{M}(\alpha) \equiv \neg\mathbf{K}(\neg\alpha)$ only holds in case of Moore's Autoepistemic Logic (i.e. $\mathbf{O}_M$), but not for Reiter's Default Logic (i.e. $\mathbf{O}_R$).

Formally, for any epistemic states e_1, e_2, world w and $z \in \mathcal{Z}$, a sentence α is true wrt. e_1, e_2, w, z, which we write as $e_1, e_2, w, z \models \alpha$, as follows:

1. $e_1, e_2, w, z \models P(n_1, \ldots, n_k)$
   iff $w[P(n_1, \ldots, n_k), z] = 1$;

[3] The third logic is Konolige's variant of Autoepistemic Logic using moderately grounded extensions [9]. For simplicity we do not consider it in this paper. Adapting our definitions and results accordingly is straightforward.

2. $e_1, e_2, w, z \models (n_1 = n_2)$
   iff n_1 and n_2 are identical standard names;
3. $e_1, e_2, w, z \models \neg\alpha$ iff $e_1, e_2, w, z \not\models \alpha$;
4. $e_1, e_2, w, z \models \alpha \wedge \beta$
   iff $e_1, e_2, w, z \models \alpha$ and $e_1, e_2, w, z \models \beta$;
5. $e_1, e_2, w, z \models \forall x\alpha$
   iff $e_1, e_2, w, z \models \alpha^x_n$ for every standard name n;

The above rules define the truth of atoms, equalities, and the usual logical connectives in the presence of standard names. The latter can be thought of as a countably infinite set of constants that satisfy the unique names assumption and an infinitary version of domain closure. Thus, first-order quantifiers can be interpreted substitutionally. Next, action modalities are defined similar as for $\mathcal{ES}$:

6. $e_1, e_2, w, z \models [n]\alpha$
   iff $e_1, e_2, w, z \cdot n \models \alpha$;
7. $e_1, e_2, w, z \models \Box\alpha$
   iff $e_1, e_2, w, z \cdot z' \models \alpha$ for all z';

Here, $z \cdot n$ refers to the result of concatenating standard name n at the end of sequence z. The meaning of belief modalities is given as follows:

8. $e_1, e_2, w, z \models \mathbf{K}(\alpha)$
   iff for every $w' \in e_1$ with $w' \simeq_z w$, $e_1, e_2, w', z \models \alpha$;
9. $e_1, e_2, w, z \models \mathbf{M}(\alpha)$
   iff for some $w' \in e_2$ with $w' \simeq_z w$, $e_1, e_2, w', z \models \alpha$;
10. $e_1, e_2, w, z \models \mathbf{O}_M(\alpha)$
   iff for every w' with $w' \simeq_z w$,
   $e_1, e_2, w', z \models \alpha$ iff $w' \in e_1$;
11. $e_1, e_2, w, z \models \mathbf{O}_R(\alpha)$
   iff for all e' with $e_1 \subseteq e'$,
   $e', e_2, w, z \models \mathbf{O}_M(\alpha)$ iff $e' = e_1$.

That is to say a sentence is known iff its true in every world of e_1, while a sentence is consistent with what is known iff some world of e_2 satisfies it. $\mathbf{O}_M$ coincides with Levesque's only-knowing (essentially the "if" in the case of $\mathbf{K}$ becomes an "iff"), while $\mathbf{O}_R$ is a variant that allows to make the connection to Reiter's Default Logic. In any case, only worlds that agree with the "real" world w on the sensing throughout z are considered, which means additional knowledge is gained by ruling out incompatible possible worlds.

We then define $e, w \models \alpha$ as $e, e, w, \langle\rangle \models \alpha$. A sentence α is *valid* (written as $\models \alpha$) iff $e, w \models \alpha$ for every e and w. A set of sentences Σ *entails* α (written as $\Sigma \models \alpha$) iff for every e, w such that $e, w \models \beta$ for all $\beta \in \Sigma$, also $e, w \models \alpha$.

Note that thus, in the above rules e_1 and e_2 are usually equal, the only exception being $\mathbf{O}_R$ where e' ranges over all possible supersets of e_1 to ensure its minimality (corresponding to a maximal set of $\mathbf{K}$-beliefs).

For $z = \langle\rangle$, the truth of subjective sentences does not depend on any world, which is why we often omit the w argument and write $e \models \alpha$ in that case. Similarly, we may write $w \models \phi$ for objective sentences ϕ. We will furthermore identify a finite set of sentences (called a *knowledge base*) with the singleton sentence given by the conjunction of all sentences in the set, i.e. we use a loose notation where a finite set can be used anywhere a formula could.

2.3 Properties

We first note the close relation of $\mathcal{ESD}$ to $\mathcal{ES}$ and $\mathcal{O}_3\mathcal{L}$:

Theorem 1. *For every sentence α without $\mathbf{O}_R$ and $\mathbf{M}$, α is valid in $\mathcal{ESD}$ iff α is valid in $\mathcal{ES}$.*

Proof. Obvious from the fact that syntax-wise, sentences without $\mathbf{O}_R$ and $\mathbf{M}$ are a subset of $\mathcal{ES}$ and the relevant semantic rules of $\mathcal{ESD}$ coincide with those of $\mathcal{ES}$. □

Theorem 2. *For every static sentence α, α is valid in $\mathcal{ESD}$ iff α is valid in $\mathcal{O}_3\mathcal{L}$.*

Proof. Obvious from the fact that syntax-wise, static sentences are a subset of $\mathcal{O}_3\mathcal{L}$ and the semantic rules of $\mathcal{ESD}$ coincide with those of $\mathcal{O}_3\mathcal{L}$ when $z = \langle\rangle$. □

We may hence directly exploit existing results for these subformalisms. If we move to the propositional case, one particularly interesting result for us here is the following relation to Reiter's Default Logic shown by Lakemeyer and Levesque [12], where a Reiter-style default of the form

$$\alpha : \beta_1, \ldots, \beta_k \,/\, \gamma \tag{4}$$

is represented by the formula

$$\mathbf{K}(\alpha) \wedge \mathbf{M}(\beta_1) \wedge \cdots \wedge \mathbf{M}(\beta_k) \supset \gamma. \tag{5}$$

While Reiter allowed for open defaults that have free variables, which then stand for the set of all their possible ground instances, we here make the restriction to propositional theories, following Lakemeyer and Levesque. A *propositional default theory* $\langle F, D\rangle$ then consists of a finite set F of static, objective, quantifier-free sentences as well as a finite set D of closed defaults of the form (4) where all of $\alpha, \beta_1, \ldots, \beta_k, \gamma$ are static, objective, and quantifier-free.

Theorem 3. *Let $\langle F, D\rangle$ be a propositional default theory, ϕ the conjunction of sentences over F, and δ the conjunction of the representations of the defaults D according to (5). Then Γ is a standard Reiter extension of $\langle F, D\rangle$ (as defined in [30]) iff there is an e such that $e \models \mathbf{O}_R(\phi \wedge \delta)$ and Γ is the set of objective beliefs of e, i.e. $\Gamma = \{\psi \mid e \models \mathbf{K}(\psi),\ \psi$ static, objective and quantifier-free$\}$.*

That is to say an epistemic state that Reiter-only-knows a knowledge base of the mentioned form corresponds exactly to an extension according to Reiter's default semantics in the sense that they believe the same objective formulas. As an important corollary, we have:

Corollary 1. *$\mathbf{O}_R(\phi \wedge \delta) \models \mathbf{K}(\psi)$ iff ψ is an element of every Reiter extension of $\langle F, D\rangle$.*

Note that there is a similar correspondence to Autoepistemic Logic when $\mathbf{O}_M$ is used instead of $\mathbf{O}_R$. The difference between the two, roughly, is how they deal with "ungrounded" expansions/extensions. While there is no default theory corresponding to only-knowing $\mathbf{K}(p) \supset p$, Autoepistemic Logic admits an expansion where p is believed.

We will also heavily rely on the following theorem of [12]. Note that while there are in general multiple possible e with $e \models \mathbf{O}_R(\Sigma_0)$ for any non-objective knowledge base Σ_0, the epistemic state that only-knows an objective knowledge base Φ is unique, and is given by the set of all worlds satisfying Φ.

Theorem 4. *Let Σ_0 be a static basic knowledge base without quantifiers. Then there is a finite set of static, objective, quantifier-free sentences $\mathfrak{E}(\Sigma_0) = \{\Phi_1, \ldots, \Phi_n\}$ such that*

$$\models \mathbf{O}_R(\Sigma_0) \equiv (\mathbf{O}_M(\Phi_1) \vee \cdots \vee \mathbf{O}_M(\Phi_n)).$$

Intuitively, if Σ_0 is of the right form (that corresponds to a default theory), $\mathfrak{E}(\Sigma_0)$ thus represents exactly the possible extensions of Σ_0.

3 BASIC ACTION THEORIES AND REGRESSION

Similar as in the classical Situation Calculus and $\mathcal{ES}$, we use basic action theories to define the agent's knowledge about the initial situation, pre- and postconditions of actions as well as sensing results:

Definition 1. *A* basic action theory *(BAT) is a set of sentences of the form* $\Sigma = \Sigma_0 \cup \Sigma_{pre} \cup \Sigma_{post} \cup \Sigma_{sense}$, *where*

1. Σ_0, the initial theory*, is a static basic knowledge base describing the initial state of the world.*
2. Σ_{pre} *is a* precondition axiom *of the form* $\Box Poss(a) \equiv \pi$*, where* π *is a fluent formula and* a *its only free variable.*[4]
3. Σ_{post} *is a finite set of* successor state axioms *(SSAs)* $\Box[a]F(\vec{x}) \equiv \gamma_F$*, one for each fluent predicate* F *relevant to the application domain, where* γ_F *is a fluent formula whose only free variables are* $\vec{x}$ *and* a*. SSAs incorporate Reiter's [31] solution to the frame problem.*[5]
4. Σ_{sense} *is a* sensing axiom *of the form* $\Box SF(a) \equiv \varphi$*, where* φ *is a fluent formula and* a *its only free variable.*

Note that while usually the initial theory Σ_0 is assumed to be objective, we here allow that it contains $\mathbf{K}$ and $\mathbf{M}$ operators in order to be able to encode defaults.

Example 1. *For the Wumpus domain, a BAT may look as follows. For simplicity, we ignore shooting arrows and grabbing the gold. The initial theory* Σ_0 *first of all contains objective formulas encoding the adjacency relation between rooms as well the initial position of the agent, which is safe:*

$$Adj(room_{11}, north, room_{12}), \ldots, Adj(room_{43}, south, room_{44})$$
$$At(room_{11}) \wedge Safe(room_{11})$$

where safety is defined as in (2). Additionally, there can be default rules such as (3) that lets the agent assume any room to be dangerous about which it is not absolutely sure. Moreover, we can use defaults to make the closed-world assumption about certain parts of the domain:

$$\mathbf{M}(\neg Adj(r, d, r')) \supset \neg Adj(r, d, r')$$

This means whenever it is safe to assume that two rooms are not connected, the agent believes that they indeed are not. Hence, the facts about Adj *above list* all and only *cases where two rooms are adjacent.*

Next, we have a precondition axiom Σ_{pre} *stating that moving in a direction* d *is possible if there is an adjacent room, and sensing breezes and stenches is always possible:*[6]

$$\Box Poss(a) \equiv \exists r, d, r'(At(r) \wedge a = move(d) \wedge Adj(r, d, r')) \vee a = sensebreeze \vee a = sensestench$$

For the At *fluent, the successor state axiom in* Σ_{post} *looks like this:*

$$\Box[a]At(x) \equiv \exists r, d(At(r) \wedge a = move(d) \wedge Adj(r, d, x)) \vee At(x) \wedge \neg\exists d\, a = move(d)$$

That is to say the position of the agent after action a *will be* x *if* a *was moving there from some adjacent room* r *by going in direction* d*, or the agent was already at* x *and did not move.*

Finally, the sensing axiom Σ_{sense} *expresses that* SF *is true for* $sensebreeze$ *iff there is a pit in an adjacent room, and true for* $sensestench$ *iff the Wumpus is nearby. For non-sensing actions such as move,* SF *will never hold:*

$$\begin{aligned}\Box SF(a) \equiv\ & a = sensebreeze \\ &\quad \wedge \exists r, d, r'(At(r) \wedge Adj(r, d, r') \wedge Pit(r')) \vee \\ & a = sensestench \\ &\quad \wedge \exists r, d, r'(At(r) \wedge Adj(r, d, r') \wedge Wumpus(r'))\end{aligned}$$

Using BATs, we can also define a regression operator for eliminating actions from formulas as is done in Reiter's Situation Calculus. For the objective case, we have the following:

Definition 2. *For any bounded, objective sentence* α *and BAT* Σ *let* $\mathcal{R}[\alpha]$, the regression of α wrt Σ*, be the fluent formula* $\mathcal{R}[\langle\rangle, \alpha]$*, where for any sequence of terms* σ *(not necessarily ground),* $\mathcal{R}[\sigma, \alpha]$ *is defined inductively on* α *by:*

1. $\mathcal{R}[\sigma, (t_1 = t_2)] = (t_1 = t_2)$*;*
2. $\mathcal{R}[\sigma, \neg\alpha] = \neg\mathcal{R}[\sigma, \alpha]$*;*
3. $\mathcal{R}[\sigma, (\alpha \wedge \beta)] = (\mathcal{R}[\sigma, \alpha] \wedge \mathcal{R}[\sigma, \beta])$*;*
4. $\mathcal{R}[\sigma, \forall x\alpha] = \forall x\mathcal{R}[\sigma, \alpha]$*;*
5. $\mathcal{R}[\sigma, [t]\alpha] = \mathcal{R}[\sigma \cdot t, \alpha]$*;*
6. $\mathcal{R}[\sigma, Poss(t)] = \mathcal{R}[\sigma, \pi^a_t]$*;*
7. $\mathcal{R}[\sigma, SF(t)] = \mathcal{R}[\sigma, \varphi^a_t]$*;*
8. $\mathcal{R}[\sigma, R(t_1, \ldots, t_k)] = R(t_1, \ldots, t_k)$ *if* R *is rigid;*
9. $\mathcal{R}[\sigma, F(t_1, \ldots, t_k)]$ *for fluent* F *is defined inductively on* σ *by:*
   (a) $\mathcal{R}[\langle\rangle, F(t_1, \ldots, t_k)] = F(t_1, \ldots, t_k))$*;*
   (b) $\mathcal{R}[\sigma \cdot t, F(t_1, \ldots, t_k)] = \mathcal{R}[\sigma, (\gamma_F)^{a\,x_1}_{t\,t_1} \cdots {}^{x_k}_{t_k}]$.

Above, π*,* φ *and* γ_F *are the right-hand sides of the corresponding axioms in* Σ_{pre}*,* Σ_{sense}*, and* Σ_{post}*, respectively.*

The general idea here is to subsequently replace formulas of the form $[t]F(\vec{t})$, $Poss(t)$ and $SF(t)$ by equivalent formulas (that do not mention actions) as defined by the BAT. Iterating such steps, a formula involving actions is thus transformed into an equivalent formula that only talks about the initial situation.

When it comes to $\mathbf{K}$ operators, consider the following theorem for $\mathcal{ES}$ which still holds in $\mathcal{ESD}$:

Theorem 5.

$$\begin{aligned}\models \Box[a]\mathbf{K}(\alpha) \equiv\ & \\ & SF(a) \wedge \mathbf{K}(SF(a) \supset [a]\alpha) \vee \\ & \neg SF(a) \wedge \mathbf{K}(\neg SF(a) \supset [a]\alpha).\end{aligned}$$

Proof. Similar as for $\mathcal{ES}$. □

The above is like a successor state axiom for the $\mathbf{K}$ operator in the sense that it relates knowing some formula after an action to the truth of a formula talking about the situation before executing that action. The difference is that this is not an axiom, but a theorem of the logic (i.e. a valid sentence). In addition, we now also have a similar one for $\mathbf{M}$:

[4] Free variables in BAT formulas are implictly assumed to be universally quantified from the outside. Furthermore, $\Box$ has lower syntactic precedence than the logical connectives, so $\Box Poss(a) \equiv \pi$ stands for $\forall a.\Box(Poss(a) \equiv \pi)$.

[5] The $[a]$ construct has higher precedence than the logical connectives, so $\Box[a]F(\vec{x}) \equiv \gamma_F$ abbreviates $\forall a.\Box(([a]F(\vec{x})) \equiv \gamma_F)$.

[6] We abuse notation here when using an action function such as $move(d)$ as our formalism does not include function symbols. Note that in the example, there are exactly four directions, so a quantified formula such as $\exists d\phi$ is to be understood as shorthand for the finite disjunction $\phi^d_n \vee \phi^d_e \vee \phi^d_s \vee \phi^d_w$, where we identify $move(n), move(e), move(s), move(w)$ with the standard names $move_n, move_e, move_s, move_w$, respectively. The restriction to standard names only is to keep the formal treatment in this paper as simple as possible and comes without loss of generality (as actions will be eliminated by means of regression); an extension by action functions as used in [3] is straightforward.

Theorem 6.

$$\models \Box[a]\mathbf{M}(\alpha) \equiv \\ \quad SF(a) \supset \mathbf{M}(SF(a) \wedge [a]\alpha) \wedge \\ \quad \neg SF(a) \supset \mathbf{M}(\neg SF(a) \wedge [a]\alpha).$$

Proof. "$\Rightarrow$": Let $e_1, e_2, w, z \models [n]\mathbf{M}(\alpha)$. Then for some $w' \in e_2$ with $w' \simeq_{z\cdot n} w$, $e_1, e_2, w', z \cdot n \models \alpha$. If $w, z \models SF(n)$, we thus have some $w' \in e_2$ with $w' \simeq_z w$, $w', z \models SF(n)$ and $e_1, e_2, w', z \models [n]\alpha$, hence $e_1, e_2, w, z \models \mathbf{M}(SF(n) \wedge [n]\alpha)$.

"$\Leftarrow$": Conversely, let $e_1, e_2, w, z \models SF(n) \supset \mathbf{M}(SF(n) \wedge [n]\alpha) \vee \neg SF(n) \supset \mathbf{M}(\neg SF(n) \wedge [n]\alpha)$ and suppose $w, z \models SF(n)$ (the other case is similar). Then $e_1, e_2, w, z \models \mathbf{M}(SF(n) \supset [n]\alpha)$, hence there is $w' \in e_2$ with $w' \simeq_z w$ such that $w', z \models SF(n)$ and $e_1, e_2, w', z \models [n]\alpha$. Therefore we have $w' \in e_2$ with $w' \simeq_{z\cdot n} w$ and $e_1, e_2, w, z \cdot n \models \alpha$, i.e. $e_1, e_2, w, z \models [n]\mathbf{M}(\alpha)$. ☐

Similarly as we use regular SSAs for regressing fluent atoms, we can therefore use the last two theorems to regress formulas involving $\mathbf{K}$ and $\mathbf{M}$. Formally, we add the following two regression rules to Definition 2:

10. $\mathcal{R}[\sigma, \mathbf{K}(\alpha)]$ is defined inductively on σ by:
    (a) $\mathcal{R}[\langle\rangle, \mathbf{K}(\alpha)] = \mathbf{K}(\mathcal{R}[\langle\rangle, \alpha])$;
    (b) $\mathcal{R}[\sigma \cdot t, \mathbf{K}(\alpha)] = \mathcal{R}[\sigma, \kappa_t^a]$,
    where κ is the right-hand side of the equivalence in Theorem 5.
11. $\mathcal{R}[\sigma, \mathbf{M}(\alpha)]$ is defined inductively on σ by:
    (a) $\mathcal{R}[\langle\rangle, \mathbf{M}(\alpha)] = \mathbf{M}(\mathcal{R}[\langle\rangle, \alpha])$;
    (b) $\mathcal{R}[\sigma \cdot t, \mathbf{M}(\alpha)] = \mathcal{R}[\sigma, \mu_t^a]$,
    where μ is the right-hand side of the equivalence in Theorem 6.

Example 2. *Consider the BAT from Example 1. The reader may verify that the following are true for regressing objective formulas:*

$$\mathcal{R}[[move(north)]At(x)] = \exists r(At(r) \wedge Adj(r, north, x))$$
$$\mathcal{R}[[a]Safe(x)] = Safe(x)$$
$$\mathcal{R}[[sensebreeze]\alpha] = \alpha$$
$$\mathcal{R}[SF(sensebreeze)] = \exists r, d, r'(At(r) \wedge Adj(r, d, r') \wedge Pit(r'))$$

That is to say after moving north the agent is at x iff x is north of the agent's previous position x. As Safe is a rigid predicate, its truth value does not change by means of any action a. Also, sensebreeze is a pure sensing action that does not have any effect on fluents, hence regressing any objective formula α through it leaves α unchanged.

Let PitNearby stand for $\exists r, d, r'(At(r) \wedge Adj(r, d, r') \wedge Pit(r'))$ and s for sensebreeze. Then we have:

$$\begin{aligned}&\mathcal{R}[\langle s\rangle, \mathbf{K}(Safe(x))]\\ &= \mathcal{R}[\langle\rangle, SF(s) \wedge \mathbf{K}(SF(s) \supset [s]Safe(x)) \vee \\ &\qquad \neg SF(s) \wedge \mathbf{K}(\neg SF(s) \supset [s]Safe(x))]\\ &= \quad PitNearby \wedge \mathbf{K}(PitNearby \supset Safe(x)) \vee \\ &\qquad \neg PitNearby \wedge \mathbf{K}(\neg PitNearby \supset Safe(x)).\end{aligned}$$

In other words the agent comes to know that room x is safe after sensing for breezes at its current location just in case there is a pit nearby and it is known that even when a pit is nearby, x is close (which can only be when x is not adjacent to the pit), or if there is no pit nearby and the agent knows that then x is safe (e.g. when the agent's position is adjacent to x and it can also rule out that x contains the Wumpus). As a next step, the above regression will then have to be checked against the agent's knowledge about the initial situation, as we will discuss in the next section. Whether or not x is actually known to be safe thus also depends on information it has gathered through previous sensing actions at various locations.

Regression is correct as given by the following theorem:

Theorem 7. *Let Σ be a BAT and α a bounded, basic sentence. Then $\mathcal{R}[\alpha]$ is static and*

1. $\mathbf{O}_R(\Sigma) \models \mathbf{K}(\alpha)$ *iff* $\mathbf{O}_R(\Sigma_0) \models \mathbf{K}(\mathcal{R}[\alpha])$.
2. $\mathbf{O}_R(\Sigma) \models \mathbf{M}(\alpha)$ *iff* $\mathbf{O}_R(\Sigma_0) \models \mathbf{M}(\mathcal{R}[\alpha])$.

Proof. Similar to the proof of Theorem 5 in [11], this follows from the fact that all transformations given by Definitions 2 and Theorems 5 and 6 are equivalence preserving wrt the BAT Σ. ☐

Hence we can reduce reasoning about knowledge and action to reasoning about knowledge in the initial situation. In order to also represent the situation where some history of actions z has already been executed, we need the following:

Definition 3. *Let $z = \langle n_1 \cdots n_k\rangle \in \mathcal{Z}$. A* sensing result *for z is a formula of the form*

$$\bigwedge_{i=1}^{k} [n_1] \cdots [n_{i-1}] \pm SF(n_i),$$

where each $\pm SF(n_i)$ is either $SF(n_i)$ or $\neg SF(n_i)$. As a special case, $\top$ is the only sensing result for $\langle\rangle$.

By incorporating sensing results through regression into the knowledge base, coming to know that α holds after executing z is equivalent to knowing initially that after z, α will come to hold:

Theorem 8. *Let Σ be a BAT, α a sentence, $z \in \mathcal{Z}$, and Ψ a sensing result for z.*

1. $\mathbf{O}_R(\Sigma) \wedge \Psi \models [z]\mathbf{K}(\alpha)$ *iff* $\mathbf{O}_R(\Sigma \wedge \mathcal{R}[\Psi]) \models \mathbf{K}([z]\alpha)$.
2. $\mathbf{O}_R(\Sigma) \wedge \Psi \models [z]\mathbf{M}(\alpha)$ *iff* $\mathbf{O}_R(\Sigma \wedge \mathcal{R}[\Psi]) \models \mathbf{M}([z]\alpha)$.

Proof. Easy to show using the fact that for evaluating both $\mathbf{K}$ and $\mathbf{M}$ operators, the semantics only considers worlds w' in the epistemic states that agree with the real world w on the sensing throughout z. ☐

Note that since $\mathcal{R}[\Psi]$ is a fluent formula, $\Sigma \wedge \mathcal{R}[\Psi]$ again conforms to our definition of a BAT as we may view $\mathcal{R}[\Psi]$ as being part of the new BAT's initial theory, i.e. we set $\Sigma_0' = \Sigma_0 \cup \{\mathcal{R}[\Psi]\}$.

4 REPRESENTATION THEOREM

To deal with knowledge, we propose a variant of Levesque and Lakemeyer's [18] Representation Theorem as follows. The general idea is to recursively replace occurrences of $\mathbf{K}(\alpha)$ by objective formulas that represent the known instances of the corresponding α according to the knowledge base Σ_0.

One important difference of our approach to Levesque and Lakemeyer's is that they directly evaluate such $\mathbf{K}(\alpha)$ formulas against the (objective) KB, whereas we defer this evaluation to a later point. Instead, we first recursively replace every $\mathbf{K}(\alpha)$ subformula in the query by a corresponding K_α, which is a special, reserved predicate not occurring in the original BAT and query, and additionally augment Σ_0 by the statement $\mathbf{K}(\alpha) \supset K_\alpha$ (and similar for $\mathbf{M}$). Intuitively, the new predicate serves as a "flag" to indicate when the corresponding belief subformula holds in an extension, and the new default rule ensures that the flag is assigned the correct value. Formally:

Definition 4. *Given a static basic formula α, $\|\alpha\|$ is defined by*

1. $\|\alpha\| = \alpha$, *when α is a static objective formula;*
2. $\|\neg\alpha\| = \neg\|\alpha\|$;
3. $\|\alpha \wedge \beta\| = \|\alpha\| \wedge \|\beta\|$;
4. $\|\exists x\alpha\| = \exists x\|\alpha\|$;
5. $\|\mathbf{K}(\alpha)\| = K_\phi(\vec{x})$, *where $\phi = \|\alpha\|$ and $\vec{x}$ are the free variables in α;*
6. $\|\mathbf{M}(\alpha)\| = M_\phi(\vec{x})$, *where $\phi = \|\alpha\|$ and $\vec{x}$ are the free variables in α.*

Moreover, $\Sigma_0 \uparrow \alpha$ is given by:

1. $\Sigma_0 \uparrow \alpha = \Sigma_0$, *when α is a static objective formula;*
2. $\Sigma_0 \uparrow \neg\alpha = \Sigma_0 \uparrow \alpha$;
3. $\Sigma_0 \uparrow \alpha \wedge \beta = \Sigma_0 \uparrow \alpha \cup \Sigma_0 \uparrow \beta$;
4. $\Sigma_0 \uparrow \exists x\alpha = \Sigma_0 \uparrow \alpha$;
5. $\Sigma_0 \uparrow \mathbf{K}(\alpha) = \Sigma_0 \uparrow \alpha \cup \{\mathbf{K}(\phi) \supset K_\phi(\vec{x})\}$, *where $\phi = \|\alpha\|$ and $\vec{x}$ are the free variables in α;*
6. $\Sigma_0 \uparrow \mathbf{M}(\alpha) = \Sigma_0 \uparrow \alpha \cup \{\mathbf{M}(\phi) \supset M_\phi(\vec{x})\}$, *where $\phi = \|\alpha\|$ and $\vec{x}$ are the free variables in α.*

We call $\|\alpha\|$ the reduction *of α and $\Sigma_0 \uparrow \alpha$ the* augmentation *of Σ_0.*

Example 3. *If $\alpha = \exists x.\, \mathbf{K}(Room(x) \wedge \neg\mathbf{K}(Pit(x))$, $\Sigma_0 \uparrow \alpha$ is Σ_0 together with the sentences*

$$\mathbf{K}(Pit(x)) \supset K_{Pit(x)}(x),$$
$$\mathbf{K}(Room(x) \wedge \neg K_{Pit(x)}(x)) \supset K_\phi(x),$$

where $\phi \stackrel{def}{=} \|Room(x) \wedge \neg\mathbf{K}(Pit(x))\| = Room(x) \wedge \neg K_{Pit(x)}(x)$. $\|\alpha\|$ then is $\exists x K_\phi(x)$.

The above construction is correct as follows. First, consider the case without nesting of $\mathbf{K}$ and $\mathbf{M}$. If e is an epistemic state such that $e \models \mathbf{O}_R(\Sigma_0)$, let

$$e \uparrow \mathbf{K}(\phi) \stackrel{def}{=} \{w \in e \mid \text{if } e \models \mathbf{K}(\phi)^{\vec{x}}_{\vec{n}}, \text{ then } w \models K_\phi(\vec{n})\}$$
$$e \uparrow \mathbf{M}(\phi) \stackrel{def}{=} \{w \in e \mid \text{if } e \models \mathbf{M}(\phi)^{\vec{x}}_{\vec{n}}, \text{ then } w \models M_\phi(\vec{n})\}$$

for objective ϕ where K_ϕ and M_ϕ do not appear in Σ_0, respectively. Intuitively, $e \uparrow \mathbf{K}(\phi)$ is like e that originally Reiter-only-knows the knowledge base Σ_0, but where in addition all its worlds also set the right values for the flag predicate associated with $\mathbf{K}(\phi)$. Then we have that $e \uparrow \mathbf{K}(\phi)$ Reiter-only-knows the augmentation (similar for $\mathbf{M}(\phi)$):

Lemma 1. *Let Σ_0 be a static basic knowledge base, ϕ an objective formula with free variables $\vec{x}$, and e, e' and e'' be epistemic states such that $e \models \mathbf{O}_R(\Sigma_0)$, $e' = e \uparrow \mathbf{K}(\phi)$ and $e'' = e \uparrow \mathbf{M}(\phi)$. Then*

1. $e \models \mathbf{O}_R(\Sigma_0)$ *iff* $e' \models \mathbf{O}_R(\Sigma_0 \uparrow \mathbf{K}(\phi))$
2. $e \models \mathbf{O}_R(\Sigma_0)$ *iff* $e'' \models \mathbf{O}_R(\Sigma_0 \uparrow \mathbf{M}(\phi))$

and for all standard names $\vec{n}$,

1. $e \models \mathbf{K}(\phi)^{\vec{x}}_{\vec{n}}$ *iff for all* $w \in e'$, $w \models K_\phi(\vec{n})$;
2. $e \models \mathbf{M}(\phi)^{\vec{x}}_{\vec{n}}$ *iff for all* $w \in e''$, $w \models M_\phi(\vec{n})$.

Proof. Since K_ϕ does not appear in Σ_0, e' behaves exactly like e except for the fact that $K_\phi(\vec{n})$ holds in all its worlds iff $\mathbf{K}(\phi)$ holds in e. Similar for e'' with $M_\phi(\vec{n})$ and $\mathbf{M}(\phi)$. ☐

Thus, we have for the recursive evaluation of formulas:

Lemma 2. *Let Σ_0 be a static basic knowledge base, α a static basic formula with free variables $\vec{x}$, and e and e' epistemic states such that $e \models \mathbf{O}_R(\Sigma_0)$ and $e' = e \uparrow \alpha$. Then $\Sigma_0 \uparrow \alpha$ is a static basic knowledge base, $\|\alpha\|$ is a static objective formula, and for all worlds $w \in e'$ and standard names $\vec{n}$,*

$$e, w \models \alpha^{\vec{x}}_{\vec{n}} \text{ iff } w \models \|\alpha\|^{\vec{x}}_{\vec{n}}.$$

Proof. This can be proven by an induction on the structure of α, using Lemma 1 for the base cases of $\mathbf{K}$ and $\mathbf{M}$ without nesting. ☐

We now exploit Theorem 4 guaranteeing that in the propositional case, Reiter-only-knowing Σ_0 boils down to only-knowing one of finitely many objective knowledge bases:

Theorem 9. *Let Σ_0 be a static basic knowledge base without quantifiers, $\mathfrak{E}(\cdots)$ as in Theorem 4, and α a static basic sentence without quantifiers. Then*

1. $\mathbf{O}_R(\Sigma_0) \models \mathbf{K}(\alpha)$ *iff for all* $\Phi \in \mathfrak{E}(\Sigma_0 \uparrow \alpha)$, $\Phi \models \|\alpha\|$.
2. $\mathbf{O}_R(\Sigma_0) \models \mathbf{M}(\alpha)$ *iff for all* $\Phi \in \mathfrak{E}(\Sigma_0 \uparrow \alpha)$, $\Phi \not\models \|\neg\alpha\|$.

Proof. This follows from Lemma 2 and Theorem 4. ☐

Thus, testing a formula $\mathbf{K}(\alpha)$ or $\mathbf{M}(\alpha)$ (possibly containing nested modalities) against a knowledge base Σ_0 (possibly containing defaults) is reducible to classical propositional entailment: First, construct the augmented knowledge base $\Sigma_0 \uparrow \alpha$ and determine its extensions. Then, for every one of the finitely many, objective extensions Φ check whether the reduced query $\|(\neg)\alpha\|$ holds in it. Note that in the propositional case, the formulas by which we augment Σ_0 can be viewed as default rules of the form (5) with either α or the β_j being empty, i.e. TRUE.

Regarding complexity, suffice it to say (without giving a formal analysis) that our Representation Theorem will not be harder than classical default reasoning (i.e., Σ_2^P-complete). The reason is that augmentation adds as many additional defaults as the query contains belief operators, but their only purpose is to set the right "flags" (values for the K_α and M_α predicates) to indicate which belief subformulas hold. The number and the internal structure of extensions remains unchanged.

5 COMPUTING EXTENSIONS

The last two sections gave us the main ingredients for implementing a knowledge-based agent whose main reasoning task during the execution of a knowledge-based program is to decide queries of the forms

$$\mathbf{O}_R(\Sigma) \wedge \Psi \models [z]\mathbf{K}(\alpha)$$
$$\mathbf{O}_R(\Sigma) \wedge \Psi \models [z]\mathbf{M}(\alpha)$$

where Ψ is the sensing result obtained for z. Theorems 8 and 7 tell us that regression can be used to incorporate sensing results into the BAT and reduce the problem to reasoning about knowledge in the initial situation only. In the propositional case that we consider here, we can then apply Theorem 9 to further reduce the problem to propositional entailment checks. The only missing piece of the puzzle now is how to determine the set $\mathfrak{E}(\cdots)$ from Theorem 4.

There are multiple options. The direct approach would be to make use of Reiter's [30] theorem that characterizes extensions by means of a fixpoint construction. Let $\langle F, D\rangle$ be the default theory corresponding to our knowledge base Σ_0, where F are the objective formulas (facts), and D the defaults corresponding to formulas of the form (5). Procedure 1 depicts the algorithm to compute extensions in pseudo code. After initializing the result to the empty set, we first determine all objective subformulas Φ in the theory (line 2). We then consider all subsets of Φ as candidates for extensions (line 3). To check whether some candidate E actually represents an extension, we start with the set of facts (line 4) and incrementally add conclusions γ for defaults whose prerequisite α holds in the previous set and whose negated justifications $\neg\beta_j$ are not in E (line 8). Since there are only finitely many defaults and we consider propositional logic, this process will eventually converge to a fixpoint (line 9). If the resulting E_i is equivalent to E (line 10), we have found an extension and include E in the result.

Procedure 1 Calculating Default Extensions

Input: a propositional default theory $\langle F, D\rangle$
Output: its set of extensions $\mathfrak{E}(\langle F, D\rangle)$

1: $\mathfrak{E}(\langle F, D\rangle) := \emptyset$;
2: $\Phi := F \cup \{\alpha, \beta_1, \ldots, \beta_k, \gamma \mid \alpha : \beta_1, \ldots, \beta_k \,/\, \gamma \in D\}$;
3: **for all** $E \subseteq \Phi$ **do**
4: $\quad E_0 := F$;
5: $\quad i := 0$;
6: $\quad$**repeat**
7: $\quad\quad i := i + 1$;
8: $\quad\quad E_i := E_{i-1} \cup \{\gamma \mid \alpha : \beta_1, \ldots, \beta_k \,/\, \gamma \in D,$ $E_{i-1} \models \alpha,\ \neg\beta_1, \ldots, \neg\beta_k \notin E\}$;
9: $\quad$**until** $E_i = E_{i-1}$;
10: $\quad$**if** for all $\phi \in \Phi$, $E \models \phi$ iff $E_i \models \phi$ **then**
11: $\quad\quad \mathfrak{E}(\langle F, D\rangle) := \mathfrak{E}(\langle F, D\rangle) \cup \{E\}$;
12: $\quad$**end if**
13: **end for**
14: **return** $\mathfrak{E}(\langle F, D\rangle)$

Note that similar to our Representation Theorem, we here again reduce the overall reasoning task to a finite number of entailment tests in classical propositional logic. Hence, in principle, all we need is a propositional reasoner (SAT solver).

However, instead of coming up with a complete re-implementation of a default reasoner, it is also possible to make use of existing off-the-shelf systems, i.e. resort to a default reasoner such as XRay [27] or DeReS [2]. Another interesting option is to employ state-of-the-art ASP solvers implementing the stable model semantics, for example `clasp`[7]. Gelfond and Lifschitz [7] showed the relation of stable models of a normal logic program with classical negation to a fragment of Reiter's Default Logic. The key idea is to identify a default

$$B_1 \wedge \cdots \wedge B_k : \overline{B_{k+1}}, \ldots, \overline{B_{k+m}} \,/\, A \tag{6}$$

with the rule

$$A \leftarrow B_1, \ldots, B_k, \mathtt{not}(B_{k+1}), \ldots, \mathtt{not}(B_{k+m}) \tag{7}$$

and a fact

$$B_1 \wedge \cdots \wedge B_k \supset A \tag{8}$$

[7] `http://www.cs.uni-potsdam.de/clasp`

with a rule

$$A \leftarrow B_1, \ldots, B_k \tag{9}$$

not containing negation as failure. Here, A and the B_i are literals and $\overline{L}$ denotes the complement of L. The correspondence then is as follows:

Theorem 10. *Let P be a program consisting of rules of the form (7) and (9), D_P the corresponding defaults (6) and F_P the corresponding facts (8). Then M is a stable model of P iff M is a maximal set of atoms such that $Th(F_P \cup M)$ is an extension of $\langle F_P, D_P\rangle$.*

Therefore, if we require Σ_0 to only contain formulas of the form (7) and (9), we can use an ASP solver to compute the extensions $\mathfrak{E}(\Sigma_0)$. Note that in general however, subformulas in facts and defaults of our knowledge base are not necessarily literals, but can be of arbitrary form, in particular when they originate from regressed sensing results $\mathcal{R}[\Psi]$ or $\mathbf{K}(\phi)$ and $\mathbf{M}(\phi)$ subformulas introduced by augmentation. One possible direction for future work is identifying syntactical restrictions on the knowledge base and queries such that the required form can be enforced. Alternatively, we can look for a way to exploit results relating SAT to ASP: Every Boolean formula ϕ can be mapped to a normal logic program P_ϕ such that the stable models of P_ϕ correspond precisely to the classical, Boolean models of ϕ, e.g. for formulas in clausal form [28] or even of arbitrary form [37].

6 RELATED WORK

The Situation Calculus and default reasoning have been combined before, albeit differently. Lee and Palla [16] present a reformulation of Situation Calculus within the stable model semantics, which allows to solve the Frame and Qualification Problems for Lin's [21] causal theories, but they do not consider sensing and knowledge. Pagnucco et al. [29] describe an account of belief change in the Situation Calculus based on Default Logic. Strass and Thielscher [38] formalize $\mathcal{D}$, a language for default reasoning about action and change and descendent of Gelfond and Lifschitz' [8] language $\mathcal{A}$, and implement it using ASP. Ryan moreover [35] presents an ASP-based interpreter for a fragment of the GOLOG programming language, but also only for the objective case.

Finally, much more work than what we have discussed here has been done on relating different non-monotonic formalisms to one another. For example, Marek and Truszczynski [24] show the correspondence between ASP and the modal logic K45. The close relation of only-knowing to classical modal systems of belief is further studied by Lakemeyer and Levesque [15]. Earlier works of integrating modal epistemic logics with default reasoning are by Lin and Shoham [23], Lifschitz [20], Amati et al. [1], and Denecker et al. [5], but either require Autoepistemic Logic to be treated differently from Default Logic or rely on a very complex semantics with fixed-point constructions.

7 CONCLUSION

We presented a formalization of knowledge-based agents with defaults based on variants of Reiter's regression and Levesque and Lakemeyer's Representation Theorem, thus laying the foundation for an implementation where backend reasoning tasks can be outsourced to off-the-shelf default reasoners.

Apart from coming up with an actual implementation and empirical evaluation thereof, there are many other possible lines for future work. One particularly promising would be to integrate our results with [14] where yet another variant of only-knowing is presented that correctly captures what it would mean to *progress* [22] a knowledge base with defaults (though only Moore-style) through an action in the presence of sensing. The formalization of [12] we have used here is problematic in that respect as it may cause inconsistencies between the default conclusions holding after an action and the facts to be forgotten in the process of progression, which is why we needed to reduce everything to the static case by means of regression. The proposed solution is to introduce additional modalities that allow to distinguish default conclusion from "hard" facts, and it would be interesting to combine our results with theirs to come up with a progression-based variant of knowledge-based agents with defaults.

ACKNOWLEDGEMENTS

This work was supported by the German Research Foundation (DFG) research unit FOR 1513 on Hybrid Reasoning for Intelligent Systems, project A1.

REFERENCES

[1] Gianni Amati, Luigia Carlucci Aiello, and Fiora Pirri, 'Definability and commonsense reasoning', *Artificial Intelligence*, **93**, 169–199, (1997).

[2] Pawel Cholewinski, V. Wiktor Marek, Miroslaw Truszczynski, and Artur Mikitiuk, 'Computing with default logic', *Artificial Intelligence*, **112**(1-2), 105–146, (1999).

[3] Jens Claßen and Gerhard Lakemeyer, 'Foundations for knowledge-based programs using $\mathcal{ES}$', in *Proceedings of the Tenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 318–328. AAAI Press, (2006).

[4] Giuseppe De Giacomo, Yves Lespérance, and Hector J. Levesque, 'ConGolog, a concurrent programming language based on the situation calculus', *Artificial Intelligence*, **121**(1–2), 109–169, (2000).

[5] Marc Denecker, V. Wiktor Marek, and Miroslaw Truszczynski, 'Uniform semantic treatment of default and autoepistemic logics', *Artificial Intelligence*, **143**(1), 79–122, (2003).

[6] Michael Gelfond, 'Answer sets', *Foundations of Artificial Intelligence*, **3**, 285–316, (2008).

[7] Michael Gelfond and Vladimir Lifschitz, 'Classical negation in logic programs and disjunctive databases', *New Generation Computing*, **9**(3/4), 365–386, (1991).

[8] Michael Gelfond and Vladimir Lifschitz, 'Representing action and change by logic programs', *Journal of Logic Programming*, **17**(2), 301–321, (1993).

[9] Kurt Konolige, 'On the relation between default and autoepistemic logic', *Artificial Intelligence*, **35**(3), 343–382, (1988).

[10] Gerhard Lakemeyer, 'The situation calculus: A case for modal logic', *Journal of Logic, Language and Information*, **19**(4), 431–450, (2010).

[11] Gerhard Lakemeyer and Hector J. Levesque, 'Situations, si! situation terms, no!', in *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, eds., Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, pp. 516–526. AAAI Press, (2004).

[12] Gerhard Lakemeyer and Hector J. Levesque, 'Only-knowing: Taking it beyond autoepistemic reasoning', in *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005)*, eds., Manuela M. Veloso and Subbarao Kambhampati, pp. 633–638. AAAI Press, (2005).

[13] Gerhard Lakemeyer and Hector J. Levesque, 'Towards an axiom system for default logic', in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*, eds., Yolanda Gil and Raymond J. Mooney, pp. 263–268. AAAI Press, (2006).

[14] Gerhard Lakemeyer and Hector J. Levesque, 'A semantical account of progression in the presence of defaults', in *Conceptual Modeling: Foundations and Applications - Essays in Honor of John Mylopoulos*, eds., Alexander Borgida, Vinay K. Chaudhri, Paolo Giorgini, and Eric S. K. Yu, volume 5600 of *Lecture Notes in Computer Science*, pp. 82–98. Springer-Verlag, (2009).

[15] Gerhard Lakemeyer and Hector J. Levesque, 'Only-knowing meets nonmonotonic modal logic', in *Proceedings of the Thirtenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*, eds., Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith. AAAI Press, (2012).

[16] Joohyung Lee and Ravi Palla, 'Reformulating the situation calculus and the event calculus in the general theory of stable models and in answer set programming', *Journal of Artificial Intelligence Research*, **43**, 571–620, (2012).

[17] Hector J. Levesque, 'All I know: a study in autoepistemic logic', *Artificial Intelligence*, **42**(2), 263–309, (1990).

[18] Hector J. Levesque and Gerhard Lakemeyer, *The Logic of Knowledge Bases*, MIT Press, 2001.

[19] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl, 'GOLOG: A logic programming language for dynamic domains', *Journal of Logic Programming*, **31**(1–3), 59–83, (1997).

[20] Vladimir Lifschitz, 'Minimal belief and negation as failure', *Artificial Intelligence*, **70**(1-2), 53–72, (1994).

[21] Fangzhen Lin, 'Embracing causality in specifying the indirect effects of actions', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 1995)*, ed., Chris S. Mellish, pp. 1985–1993. Morgan Kaufmann Publishers Inc., (1995).

[22] Fangzhen Lin and Raymond Reiter, 'How to progress a database', *Artificial Intelligence*, **92**(1–2), 131–167, (1997).

[23] Fangzhen Lin and Yoav Shoham, 'Epistemic semantics for fixed-points nonmonotonic logics', in *Proceedings of the Third Conference on Theoretical Aspects on Reasoning about Knowledge (TARK 1990)*, ed., Rohit Parikh, pp. 111–120. Morgan Kaufmann Publishers Inc., (1990).

[24] Wiktor W. Marek and Miroslaw Truszczynski, *Nonmonotonic logic: context-dependent reasoning*, Springer-Verlag, 1993.

[25] John McCarthy and Patrick Hayes, 'Some philosophical problems from the standpoint of artificial intelligence', in *Machine Intelligence 4*, eds., B. Meltzer and D. Michie, 463–502, American Elsevier, New York, (1969).

[26] Robert C. Moore, 'Semantical considerations on nonmonotonic logic', *Artificial Intelligence*, **25**(1), 75–94, (1985).

[27] Pascal Nicolas and Torsten Schaub, 'The XRay system: An implementation platform for local query-answering in default logics', in *Applications of Uncertainty Formalisms*, eds., Anthony Hunter and Simon Parsons, volume 1455 of *Lecture Notes in Computer Science*, pp. 354–378. Springer-Verlag, (1998).

[28] Ilkka Niemelä, 'Logic programs with stable model semantics as a constraint programming paradigm', *Annals of Mathematics and Artificial Intelligence*, **25**(3–4), 241–273, (1999).

[29] Maurice Pagnucco, David Rajaratnam, Hannes Strass, and Michael Thielscher, 'Implementing belief change in the situation calculus and an application', in *Proceedings of the Twelfth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2013)*, eds., Pedro Cabalar and Tran Cao Son, volume 8148 of *Lecture Notes in Computer Science*, pp. 439–451. Springer-Verlag, (2013).

[30] Raymond Reiter, 'A logic for default reasoning', *Artificial Intelligence*, **13**(1–2), 81–132, (1980).

[31] Raymond Reiter, 'The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression', *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, 359–380, (1991).

[32] Raymond Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001.

[33] Raymond Reiter, 'On knowledge-based programming with sensing in the situation calculus', *ACM Transactions on Computational Logic*, **2**(4), 433–457, (2001).

[34] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 3rd edn., 2009.

[35] Malcolm Ryan, 'Efficiently implementing GOLOG with answer set programming', in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*, eds., Carla E. Brodley and Peter Stone, pp. 2352–2357. AAAI Press, (2014).

[36] Richard B. Scherl and Hector J. Levesque, 'Knowledge, action, and the frame problem', *Artificial Intelligence*, **144**(1–2), 1–39, (2003).

[37] Igor Stéphan, Benoit Da Mota, and Pascal Nicolas, 'From (quantified) boolean formulae to answer set programming', *Journal of Logic and Computation*, **19**(4), 565–590, (2009).

[38] Hannes Strass and Michael Thielscher, 'A language for default reason-

ing about actions', in *Correct Reasoning - Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, eds., Esra Erdem, Joohyung Lee, Yuliya Lierler, and David Pearce, volume 7265 of *Lecture Notes in Computer Science*, pp. 527–542. Springer-Verlag, (2012).