Practical reasoning about knowledge states for open world planning with sensing

Tamara Babaian [*] — **James G. Schmolze** [**]

[*] *Department of Computer Information Systems*
Bentley College
Waltham, MA 02452 (USA)
tbabaian@bentley.edu

[**] *Department of Computer Science*
Tufts University
Medford, MA 02155 (USA)

ABSTRACT. *We present a representation for reasoning and planning with an incomplete state description (open-world) called PSIPLAN-S. The presented formalism has several properties critical for application domains with a large degree of incompleteness in the state description, particularly, in domains with a large or unknown set of all objects. The formalism offers (1) considerably expressive state and goal description language, that includes limited universal quantification, (2) representation of sensing actions and knowledge goals, (3) a correct and complete state update procedure, and (4) complete reasoning within a substantial subset of the language. The approach is illustrated by examples from a working system.*

KEYWORDS: *automated planning, planning with incomplete information and sensing, reasoning about actions.*

1. Introduction

Planning with correct but incomplete information and with or without sensing actions has been addressed by numerous researchers, offering several approaches for knowledge representation and reasoning with an incomplete state description (e.g., (Scherl *et al.*, 1993; Moore, 1985; Etzioni *et al.*, 1997; Petrick *et al.*, 2002; Son *et al.*, 2001; Levesque *et al.*, 1997; Eiter *et al.*, 2004; Liu *et al.*, 2005; Thielscher, 2005)).

One major challenge of open-world planning is the computational complexity of reasoning with incomplete information. The PSIPLAN-S framework, which we present in this paper, occupies a unique niche by offering: (1) a logic of knowledge that has sufficient expressive power for a large array of applications, (2) polynomial time algorithms to determine entailment and other key relationships for a large subset of the language, and (3) completeness of reasoning for a large subset of the language – and we conjecture that it is complete for the entire language. We report on the logical foundations of this representation and give examples of its important properties.

We assume that the agent's knowledge is correct but incomplete and that it has a set of primitive actions available. We assume that the set of all domain objects is unknown and infinite. *Domain* actions change the world, have deterministic effects and do not have conditional effects. *Sensing* actions do not change the state of the world, but instead return information.

Our representation can be used with a variety of planning languages and planning algorithms. To date it has been used for a conformant (i.e., with incomplete information but without sensing) partial order planner (Babaian *et al.*, 2000), a conformant Graphplan-based planner (Carlin *et al.*, 2005), and a planner that interleaves planning and execution (Babaian *et al.*, 2002). At the core of any planning language and algorithm is the need to project or regress an agent's state of knowledge over a sequence of primitive actions. For this paper, we focus only on this core and on progression, and therefore define a planning problem to be a 3-tuple $\langle I, G, A\rangle$ where I is an initial knowledge base of the agent, G is a goal to achieve, described as a partial knowledge state, and A is a set of primitive actions. A plan P is an *executable* sequence of ground instances of elements of A. To be *executable* means that it can be proven that the agent will know each action's preconditions are true just before executing the action. Finally, a plan *solves* a planning problem if and only if it can be proven that if I is true before the first action then G is true after the last.

Parts of PSIPLAN-S have been presented in (Babaian *et al.*, 2002; Babaian *et al.*, 2000), both of which addressed the use of PSIPLAN-S representation in planning algorithms. (Babaian *et al.*, 2006) focuses on the calculus and algorithms behind PSIPLAN - a subset of PSIPLAN-S that does not handle sensing actions and knowledge goals (conformant planning). In this paper we focus on the logical foundations of reasoning and planning with sensing actions and knowledge goals and prove properties underlying its soundness and completeness that have not been published before.

***Overview of* PSIPLAN-S**

To illustrate the representational needs of an open-world planner that are addressed in PSIPLAN-S we present the following scenario from a real-world application called Writer's Aid (Babaian *et al.*, 2002). An agent must locate and possibly download papers that contain certain keywords in their title. The agent knows nothing about any actual papers, their locations or contents. It knows about a few bibliographic websites and it can execute actions for querying these sites and downloading papers from a given location.

In order to complete its task the agent must (1) find all relevant papers from these bibliographies, (2) determine if these bibliographic collections have viewable versions of the papers, and (3) if a paper is located, download it; otherwise, try another bibliographic source.

First, it must be possible to distinguish between those bibliographies known to be preferred by the user, known not to be preferred, and those about which the agent does not know the user's preference. The closed world assumption cannot be used here, so negative facts must be included in the state description. When the number of domain individuals is large or unknown, as it is in Writer's Aid due to the number of potential bibliographies and citable papers, it is impossible to represent the known negative facts without some sort of universal quantification. Moreover, when the domain is either infinite or not entirely known by the agent, reasoning with quantified propositions cannot be done by expanding each to the set of ground instances it represents.

PSIPLAN-S uses propositions called ψ-forms to represent such quantified statements and introduces a calculus of operations on ψ-forms to support automated reasoning. Formal definitions are presented in Section 3. Here we informally introduce ψ-forms and discuss their use in reasoning and planning with incomplete information.

Briefly, a ψ-form $[Q(\vec{x})$ except $\{\sigma_1, \ldots, \sigma_n\}]$ represents a set of all ground instances of $Q(\vec{x})$ except those that can be obtained using one of the substitutions $\sigma_1, \ldots, \sigma_n$, which are substitutions on a subset of variables of $\vec{x}$. For example,

$$\psi_1 = [Kn(\neg \mathit{PrefBib}(b)) \text{ except } \{\{b = ACM\}, \{b = RI\}\}] \quad (1)$$

denotes all ground instances of the formula $Kn(\neg \mathit{PrefBib}(b))$ minus two exceptions: $Kn(\neg \mathit{PrefBib}(ACM))$ and $Kn(\neg \mathit{PrefBib}(RI))$, and could also be expressed as a universally quantified formula

$$\forall b \,.\, Kn(\neg \mathit{PrefBib}(b)) \vee (b = ACM) \vee (b = RI).$$

PSIPLAN-S uses a combination of ground atoms and ψ-forms to represent incomplete information. For example, to represent statement *ST* below

> *ST*: The only preferred bibliographies are the digital library of the ACM and possibly the ResearchIndex database.

PSIPLAN-S uses two propositions (2) (see below) and (1) which taken *together*, express *ST*:

– a *Kn*-atom, expressing *"It is known that ACM's digital library is a preferred bibliography":*

$$Kn(PrefBib(ACM)) \tag{2}$$

– ψ-form ψ_1 defined above in (1), expressing *"It is known that nothing is a preferred bibliography except for possibly ACM and the ResearchIndex."*
Note that ψ_1 alone does not commit to the truth or falsity of its exception clauses $Kn(\neg PrefBib(ACM))$ and $Kn(\neg PrefBib(RI))$, but from (2) and (1) we can conclude that the agent knows that ACM is a preferred bibliography, and nothing else is a preferred bibliography except for possibly the ResearchIndex.

Note that while PSIPLAN-S can represent infinite "negative" knowledge via ψ-forms, it can represent only finite "positive knowledge", in other words a finite number of propositions of the form $Kn(a)$. This design reflects the assumption typically made in all closed wold problems, that in a planning problem there is a finite number of relevant things that are true, and maybe an infinite number of things that are false. For instance, there is only a finite number of items in any briefcase, while there is an infinite number of objects that are not in it.

Second, it must be possible to represent *knowledge goals*[1], like *identify the relevant papers recommended by the digital library of the ACM*, and infer that such goals can be achieved by using a sensing action, which queries a bibliography for a list of papers related to a keyword. PSIPLAN-S uses *KW*-propositions to represent *knowing whether* a statement is true or false. For example, the goal above, as well as the effect of the sensing action is expressed with the following *KW*-ψ-form

$$\tilde{\psi}_2 = [KW(\neg InCollection(y, ACM) \vee \neg Rel(y, Kwrd))], \tag{3}$$

which states that for every possible paper y, the agent knows whether this paper is in the collection of ACM and is also recommended as related to keyword $Kwrd$. Note that $KW(p)$ is a shortcut for $Kn(p) \vee Kn(\neg p)$, and therefore $KW(p)$ is equivalent to $KW(\neg p)$. For the convenience of algorithm formulation, we use the disjunction of negated literals inside of the $KW()$ operator instead of an equivalent conjunction of atoms.

Third, it must be possible to download all relevant papers. Thus the agent must reason about *domain goals*, such as having a paper stored on a hard drive. PSIPLAN-S's domain goals include atoms and universally quantified propositions represented with ψ-forms.

Furthermore, for reasons of efficiency, the agent should avoid redundant actions. There is no need to download a paper that is known to be stored on the local hard drive,

1. It would be more accurate to call them *knowing-whether* goals, however the term *knowledge goal* has been used in the related planning literature to refer to the same concept.

and no need to sense for information already implied by the agent's knowledge base. Thus the agent needs to know what it does and does not know. PSIPLAN-S provides complete reasoning from the agent's state of knowledge about domain and knowledge goals. One of the features distinguishing PSIPLAN-S from other implemented planning formalisms is the fact that the reasoning algorithms work directly with quantified propositions and never convert them to sets of ground instances.

Another aspect contributing to non-redundancy of information gathering is the knowledge update procedure of PSIPLAN-S's, which produces a knowledge state that correctly and completely describes the set of possible worlds resulting from taking an action. This is done without actually considering all possible worlds individually. Although we do not report on it in this paper, the reasoning from an agent's state of knowledge and the state update procedure have polynomial complexity in the number and size of propositions in the agent's state of knowledge (Babaian *et al.*, 2006).

PSIPLAN-S does not allow conditional effects and effects with universal quantification of atoms, commonly handled by many other planners (e.g. (Tu *et al.*, 2007; Golden, 1998)). These omissions are deliberate, since they allow PSIPLAN-S to specify the effects of the domain actions precisely, which is important for ensuring the computational properties of the knowledge updates in PSIPLAN-S, in particular, for keeping the knowledge updates correct, complete and polynomial time complex in the number of participating propositions. These properties of the state update bear on the properties of the planning algorithms, such as soundness and completeness of conformant planning and avoiding redundancy in using sensing and domain actions. We discuss the impact of conditional effects on the complexity of planning with incomplete information in (Babaian *et al.*, 2006).

To illustrate the difference between PSIPLAN-S and other formalisms, consider the effect of emptying a box. Such an effect can be described in a conditional form as "if object x was inside the box, then x would be outside of it". Alternatively, this effect can be described as a universally quantified statement "all objects will be outside of the box". Formulated in this way, the effect asserts what is *true* upon executing the action, but does not necessarily designate what exactly has *changed* in the world. PSIPLAN-S does not allow conditional effects and universal quantification over atoms, but as a solution to this problem it is possible to define an action of removing a single object $?x$[2] from the box. The action would have a precondition "object $?x$ is inside the box" and an atomic effect "object $?x$ is not inside the box", which describes precisely the changes to the world caused by the action. While this representation is not ideal, it allows to approximate the action of emptying the box with a series of individual object removals. The benefits of the action language used in PSIPLAN-S are the computational properties that we described above.

Another difference between PSIPLAN-S and other formalisms is the absence of universally quantified *positive implication* goals, such as "all book objects are inside box A" (in other words, $\forall x.Kn(Book(x) \supset In(BoxA, x))$. These goals are not part

2. note that we use $?x$ to denote a parameter of an action, versus x, which denotes a variable

of PSIPLAN-S, because they cannot be handled naturally by its logical mechanisms. However, they can be handled if planning is interleaved with execution of partial plans, as done in Writer's Aid (Babaian *et al.*, 2002), by first achieving the goal of knowing the identity of all book objects, i.e. $[KW(Book(x))]$ and then posting individual goals for all identified book objects.

2. Representing an agent's knowledge

Before defining the language of PSIPLAN-S formally, we define two languages that do not have quantification, $\mathcal{L}_\mathcal{D}$ and $\mathcal{L}_\mathcal{K}$. $\mathcal{L}_\mathcal{D}$ is a ground *domain* language, used to define the state of the world. $\mathcal{L}_\mathcal{K}$ is a ground *knowledge* language, expressing the agent's knowledge of the world state in the form of ground knowledge propositions. PSIPLAN-S is an open world planning language, which includes

1) a compact representation of propositions of $\mathcal{L}_\mathcal{K}$ used to denote agent's knowledge of the world state and goals;
2) an action definition language.

$\mathcal{L}_\mathcal{D}$ and $\mathcal{L}_\mathcal{K}$ are not directly used in planning, because our goal is to provide a planning representation which could operate in domains where the number of objects is infinite or at least very large, and can never be fully acquired. Thus, it is impossible to enumerate all ground facts using $\mathcal{L}_\mathcal{D}$ or $\mathcal{L}_\mathcal{K}$.

However, we introduce $\mathcal{L}_\mathcal{D}$ and $\mathcal{L}_\mathcal{K}$ for the purpose of defining PSIPLAN-S's semantics and proving some of its important logical properties in the rest of this section. We adopt the following standard definitions throughout the paper.

DEFINITION. — *A set s of propositions is called* satisfiable*, if there exists a model of all propositions in s. A set s of propositions is called* consistent*, if there exists a proposition p such that $s \Rightarrow p$ is not provable using the inference rules of the language. Otherwise, s is called* inconsistent*.*

Furthermore, for $\mathcal{L}_\mathcal{D}$, $\mathcal{L}_\mathcal{K}$ and PSIPLAN-S we assume an infinite number of domain constants, including propositional symbols, which denote distinct domain individuals. We use the unique names assumption on the constants and propositional symbols of the languages we define.

2.1. *The propositional domain language $\mathcal{L}_\mathcal{D}$*

DEFINITION (DOMAIN LANGUAGE $\mathcal{L}_\mathcal{D}$). — *The alphabet of $\mathcal{L}_\mathcal{D}$ consists of a countable set of propositional symbols and logical connectives $\neg$ and $\vee$.*

Assume $a, a_1, \ldots, a_n$ are propositional symbols. A formula *in $\mathcal{L}_\mathcal{D}$ (also called a* domain proposition *) is either*

1) a, called a domain atom *or*

2) $\neg a_1 \vee \neg a_2 \vee \ldots \vee \neg a_n$, *where* $n \geq 1$, *called a* domain clause.

DEFINITION. — *Domain propositions are interpreted over a set of* worlds, *which are standard propositional interpretations. For a world* w *and a domain proposition* p, $w(p)$ *denotes that* p *is* $true$ *in the world* w *according to the standard interpretation rules of propositional formulas. Such world* w *is then called a* model *of* p.

Assuming standard propositional semantics, it is easy to see that deduction from a satisfiable set of domain propositions in $\mathcal{L}_{\mathcal{D}}$ is complete under unit clause resolution and subsumption:

$$(r1) \quad \frac{a;\ \neg a \vee \neg a_1 \vee \ldots \vee \neg a_k}{\neg a_1 \vee \ldots \vee \neg a_k} \text{ (unit clause resolution),}$$

$$(r2) \quad \frac{\neg a_1 \vee \ldots \vee \neg a_k}{\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k} \text{ (subsumption).}$$

To extend the completeness result to any set of propositions from $\mathcal{L}_{\mathcal{D}}$, we only need to add the third rule

$$(r3) \quad \frac{a, \neg a}{p} \text{ , where } p \text{ is any domain proposition.}$$

2.2. *The propositional SOK language* $\mathcal{L}_{\mathcal{K}}$

We define $\mathcal{L}_{\mathcal{K}}$, a language for representing the state of agent's knowledge of the world and of its own knowledge. The formulas of $\mathcal{L}_{\mathcal{K}}$, are called *knowledge propositions* as defined below.

DEFINITION (SOK LANGUAGE $\mathcal{L}_{\mathcal{K}}$). — *The alphabet of* $\mathcal{L}_{\mathcal{K}}$ *consists of modal symbols* $Kn()$ *and* $KW()$, *and the alphabet of* $\mathcal{L}_{\mathcal{D}}$.

Assume p *denotes a domain proposition, i.e.* $p \in \mathcal{L}_{\mathcal{D}}$. *A* formula *in* $\mathcal{L}_{\mathcal{K}}$ *(also called a* knowledge proposition*) is either*

1) $Kn(p)$, *called a* a *Kn*-atom,
2) $\neg Kn(p)$,
3) $KW(p)$, *called a* a *KW*-atom, *or*
4) $\neg KW(p)$.

Kn-atoms represent the agent's knowledge of the world, e.g. $Kn(On(A, B))$ represents that the agent knows that block A is on block B.

$KW(p)$ denotes that the agent knows the truth value of p, and is thus semantically equivalent to $Kn(p) \vee Kn(\neg p)$ (note that this disjunctive formula is outside of $\mathcal{L}_{\mathcal{K}}$).

2.3. *Semantics and entailment*

We use *k-states* of Baral and Son (Son *et al.*, 2001) to define the semantics of the knowledge language $\mathcal{L}_{\mathcal{K}}$. Let $\mathcal{W}$ denote the set of all worlds, which are propositional interpretations over a given domain of discourse.

DEFINITION. — *A* k-state *is a pair* (w, W)*, where* W *is a subset of* $\mathcal{W}$ *and* w *is an element of* W*. A k-state* (w, W) *represents the knowledge state of an agent who actually being in the world* w *thinks it might be in any of the worlds of* W*. We are assuming that the agent's knowledge is* correct*, hence we* require *that for any k-state* (w, W), $w \in W$*. The set of all possible k-states is denoted* $\mathcal{K}$*.*

The set of *models* of a knowledge proposition or a set of knowledge propositions from $\mathcal{L}_{\mathcal{K}}$ is denoted by $\alpha()$ and defined below.

DEFINITION. — *For any domain proposition* $p \in \mathcal{L}_{\mathcal{D}}$*:*

1) $\alpha(Kn(p)) = \{(w, W) \mid w \in W \wedge \forall w' \in W \,.\, w'(p)\}$
2) $\alpha(\neg Kn(p)) = \{(w, W) \mid w \in W \wedge \exists w' \in W \,.\, w'(\neg p)\}$
3) $\alpha(KW(p)) = \{(w, W) \mid w \in W \wedge [\forall w' \in W \,.\, w'(p)] \ \vee \ [\forall w' \in W \,.\, w'(\neg p)]\}$
4) $\alpha(\neg KW(p)) = \{(w, W) \mid w \in W \wedge [\exists w' \in W \,.\, w'(p)] \wedge [\exists w'' \in W \,.\, w''(\neg p)]\}$.

In the above, and throughout this paper, assuming $p = \neg a_1 \vee \ldots \vee \neg a_k$*, we use* $w(\neg p)$ *as a shorthand for* $w(a_1), \ldots, w(a_k)$*.*

Note that according to the definition above, for each $k \in \mathcal{L}_{\mathcal{K}}$, we have $\alpha(\neg k) = \mathcal{K} - \alpha(k)$.

DEFINITION. — *The set of models of a set of knowledge propositions* $\{k_1, \ldots, k_m\}$ *is the intersection of the models of all* k_i*s , i.e.* $\alpha(\{k_1, \ldots, k_m\}) = \cap_{i=1}^{m} \alpha(k_i)$*.*

DEFINITION (ENTAILMENT IN $\mathcal{L}_{\mathcal{K}}$, $\models_k$). — *For two knowledge propositions or sets of knowledge propositions* q *and* r *we write* $q \models_k r$ *if and only if every model of* q *is a model of* r*, i.e.*

$$q \models_k r \text{ if and only if } \alpha(q) \subseteq \alpha(r) \tag{4}$$

The following Proposition states an important property of $\mathcal{L}_{\mathcal{K}}$ that allows the reduction of the entailment between *Kn*-propositions to the usual propositional entailment. In particular, determining satisfiability of a set *Kn*-atoms is reduced to the propositional case. It also proves that in our logic an agent *knows all the propositional consequences of its knowledge*.

PROPOSITION 1. — *Let* $p_1, \ldots, p_m, q$ *be domain propositions:*

$$(p_1, \ldots, p_m \models q) \ \Leftrightarrow \ (Kn(p_1), \ldots, Kn(p_m) \models_k Kn(q)).$$

Some special cases of entailment between propositions of our logic are considered in the two propositions below. These properties are later used in establishing the soundness and completeness properties of PSIPLAN-S. Proposition 2 considers entailment between a set of *Kn*-atoms and a single *KW*-atom. Proposition 3 considers entailment between *KW*-atoms.

Please note that in Proposition 2 and further in the paper, we often treat disjunctive clauses as sets of literals and use set notation to describe relationships between such clauses.

PROPOSITION 2. — *Let* c *be a domain clause.*

1) If a *is a domain atom,* $Kn(a) \models_k KW(c)$ *if and only if* $c = \neg a$.

2) If c' *is a domain clause* $Kn(c') \models_k KW(c)$ *if and only if* $c' \subseteq c$.

3) If $a_1, \ldots, a_r$ *are domain atoms, and* $c_1, \ldots, c_n$ *are domain clauses, and the set* $\{a_1, \ldots, a_r, c_1, \ldots, c_n\}$ *is consistent,*

$$s = \{Kn(a_1), \ldots, Kn(a_r), Kn(c_1), \ldots, Kn(c_n)\} \models_k KW(c)$$

if and only if

a) there exists a set of domain atoms $a_{i_1}, \ldots, a_{i_m}$*, such that* $Kn(a_{i_j}) \in s$ *and* $c = \neg a_{i_1} \vee \ldots \vee \neg a_{i_m}$*, or*

b) there exists a domain clause c'*, such that* $s \models_k Kn(c')$ *and* $c' \subseteq c$.

PROPOSITION 3. — $KW(c_1), \ldots, KW(c_n) \models_k KW(c)$ *if and only if there exists a set of indices* $i_1, \ldots, i_m$ *where* $1 \leq i_j \leq n$ *for each* i_j*, such that* $c = c_{i_1} \vee \ldots \vee c_{i_m}$.

In other worlds, for a set of KW-atoms to entail another KW-atom $KW(c)$, there must be a subset whose domain clauses produce the domain clause c when combined via the union operator. Note that when the left side consists of just a single KW-atom $KW(c')$, we have $KW(c') \models_k KW(c)$ if and only if $c' = c$.

2.4. *Inference rules*

The soundness of inference rules depicted in Figure 1 can be easily verified using the definition of entailment and Propositions 1-3. We conjecture, but have not shown, that this set of rules is complete for $\mathcal{L}_{\mathcal{K}}$.

The two propositions presented later in this section establish that the rule system is complete with respect to reasoning about *Kn* and *KW*-atoms from a satisfiable set of *Kn*-atoms. As follows from Proposition 1, satisfiability of a set of *Kn*-atoms can be established by checking the consistency of the set of corresponding domain propositions. This result is essential for proving the completeness of reasoning and planning without sensing in PSIPLAN-S, since, as we define in the next section, the state of agent's knowledge is always described with a satisfiable set of *Kn*-atoms.

PROPOSITION 4. — *Suppose* $s = \{Kn(p_1), \ldots, Kn(p_m)\}$ *is a satisfiable set.* $s \models_k Kn(c)$ *if and only if* $s \Rightarrow Kn(c)$ *can be derived using rules (R1) and (R2).*

(R1) $$\frac{Kn(a), Kn(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k)}{Kn(\neg a_1 \vee \ldots \vee \neg a_k)}$$

(R2) $$\frac{Kn(\neg a_1 \vee \ldots \vee \neg a_k)}{Kn(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k)}$$

(R3) $$\frac{Kn(a)}{KW(\neg a)}\ ,\ \frac{Kn(\neg a)}{KW(\neg a)}$$

(R4) $$\frac{Kn(\neg a_1 \vee \ldots \vee \neg a_k)}{KW(\neg a_1 \vee \ldots \vee \neg a_k)}$$

(R5) $$\frac{KW(\neg a_1 \vee \ldots \vee \neg a_k), KW(\neg a_1' \vee \ldots \vee \neg a_m')}{KW(\neg a_1 \vee \ldots \vee \neg a_k \vee \neg a_1' \vee \ldots \vee \neg a_m')}$$

(R6) $$\frac{KW(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k), Kn(a)}{KW(\neg a_1 \vee \ldots \vee \neg a_k)}$$

(R7) $$\frac{KW(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k), \neg Kn(a)}{Kn(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_k)}$$

(R8) $$\frac{KW(\neg a_1 \vee ... \vee \neg a_k), \neg Kn(\neg a_1 \vee ... \vee \neg a_k)}{Kn(a_1), ..., Kn(a_k)}$$

(R9) $$\frac{Kn(a)}{\neg Kn(\neg a)}\ ,\ \frac{Kn(\neg a)}{\neg Kn(a)}$$

(R10) $$\frac{Kn(a_1), \ldots, Kn(a_k)}{\neg Kn(\neg a_1 \vee \ldots \vee \neg a_k)}$$

(R11) $$\frac{\neg KW(\neg a)}{\neg Kn(\neg a), \neg Kn(a)}$$

(R12) $$\frac{\neg Kn(\neg a), \neg Kn(a)}{\neg KW(\neg a)}$$

(R13) $$\frac{\neg Kn(a_1), \neg Kn(\neg a_1 \vee \ldots \vee \neg a_k)}{\neg KW(\neg a_1 \vee \ldots \vee \neg a_k)}$$

(R14) $$\frac{Kn(a), Kn(\neg a)}{Kn(p)}$$, for any $p \in \mathcal{L}_{\mathcal{D}}$,

(R15) $$\frac{KW(\neg a_1 \vee \ldots \vee \neg a_n), \neg KW(\neg a_1 \vee \ldots \vee \neg a_n)}{Kn(p)}$$, for any $p \in \mathcal{L}_{\mathcal{D}}$,

(R16) $$\frac{Kn(\neg a_1 \vee \ldots \vee \neg a_n), \neg Kn(\neg a_1 \vee \ldots \vee \neg a_n)}{Kn(p)}$$, for any $p \in \mathcal{L}_{\mathcal{D}}$,

Figure 1. *Inference rules for* $\mathcal{L}_{\mathcal{K}}$

PROPOSITION 5. — *Suppose* $s = \{Kn(p_1), \ldots, Kn(p_m)\}$ *is a satisfiable set.* $s \models_k KW(c)$ *if and only if* $s \Rightarrow KW(c)$ *can be derived using rules (R1), (R2), (R3), (R4) and (R5).*

The significance of the Conjecture below is that allows to limit the number of rules used by PSIPLAN-S reasoning with sensing to only 6, while the completeness of such reasoning would be retained if the conjecture holds.

This is possible because of a number of assumptions and choices made in PSIPLAN-S. For instance, the syntax of PSIPLAN-S action and state descriptions, the requirement on the correctness of agent's knowledge, no information loss and complete and precise specification of action effects allow PSIPLAN-S to operate without the rules that use negated *Kn*-propositions. We have presented all rules here to provide a more complete description of the propositional language $\mathcal{L}_{\mathcal{K}}$, which can be used in formalisms that do not make some of the assumptions made in PSIPLAN-S and thus may need to rely on another subset of these rules.

CONJECTURE. — Assume $s = \{Kn(p_1), \ldots, Kn(p_m), KW(q_1), \ldots, KW(q_l)\}$ is a satisfiable set.

– $s \models_k Kn(r)$ if and only if $s \Rightarrow Kn(r)$ can be derived using rules (R1) and (R2).
– $s \models_k KW(r)$ if and only if $s \Rightarrow KW(r)$ can be derived using rules (R1), (R2), (R3), (R4), (R5) and (R6).

□

2.5. *The agent's state of knowledge (SOK)*

In this section we introduce other definitions and properties required in the context of planning.

DEFINITION. — *An agent's state of knowledge, or* SOK*, is a satisfiable set of* Kn *atoms. The set of* possible worlds *corresponding to SOK* s *is denoted* $\mathcal{B}(s)$ *and defined as the set of all worlds in all models of* s*:*

$$\mathcal{B}(s) = \{w \,|\, \exists (w, W) \in \alpha(s)\}.$$

PROPOSITION 6. — $\mathcal{B}(s) = \{w \,|\, \forall p \in \mathcal{L}_{\mathcal{D}} \,.\, (s \models_k Kn(p)) \Rightarrow w(p)\}$*, or in other words, the set of possible worlds is equivalent to the set of all worlds in which everything known to the agent is true.*

We make the *Closed Knows Whether Assumption*, which means that for a given SOK s and for all domain propositions p, if s does not entail $KW(p)$, i.e., $s \not\models_k KW(p)$, then the agent assumes that $\neg KW(p)$.

DEFINITION. — *For a SOK* s *we define*

$$CKWA(s) = s \cup \{\neg KW(p) \,|\, p \in \mathcal{L}_{\mathcal{D}} \wedge s \not\models_k KW(p)\}.$$

The set of possible worlds of s and of its closure $CKWA(s)$ are the same, as shown next. This allows to eliminate the direct use of the negated *KW*-propositions in PSIPLAN-S relying on the negation as failure instead: if $KW(p)$ cannot be established via application of rules identified in Conjecture 1, we assume ignorance about p, i.e. $\neg KW(p)$.

PROPOSITION 7. — $\mathcal{B}(s) = \mathcal{B}(CKWA(s))$

3. ψ-forms - a compact representation

We now introduce propositions called ψ-forms that compactly represent sets of $\mathcal{L}_{\mathcal{K}}$ propositions. We define ψ-forms before introducing PSIPLAN-S fully in Section 4.

We note that in ψ-forms the *KW* operator is used only with a negated clause. Using *KW*-propositions with a conjunction of atoms inside would result in more cumbersome formulations of the theorems, while giving us no additional expressive power, since $KW(\neg a_1 \vee \ldots \vee \neg a_m)$, is equivalent to $KW(a_1 \wedge \ldots \wedge a_m)$.

DEFINITION. — *The general form of a ψ-form is $[Q(\vec{x})$ except $\{\sigma_1, \ldots, \sigma_n\}]$. The formula $Q(\vec{x})$ is called the* main form *and has the form $Kn(p(\vec{x}))$ or $KW(p(\vec{x}))$ where $p(\vec{x})$ is a disjunction of negated literals. The main form of ψ is denoted $\mathcal{M}(\psi)$. Each of $\sigma_1, \ldots, \sigma_n$ defines ψ's* exceptions*, and each exception is a substitution on a subset of variables in $\vec{x}$.*

A ψ-form may have an empty set of exceptions. In that case it has a form $[Q(\vec{x})]$ and is called simple.

ψ-forms represent the set of all ground instantiations of the main form, except for those that can be obtained by instantiating the exceptions, as illustrated by (1) earlier and defined below.

DEFINITION. — *The set of ground propositions (instances) of $\mathcal{L}_{\mathcal{K}}$ represented by a ψ-form is defined using the Inst function, where*

1) $Inst([Q(\vec{x})]) = \{Q(\vec{x})\sigma \mid Q(\vec{x})\sigma \text{ is ground}\}$

2) $Inst([Q(\vec{x}) \text{ except } \{\sigma_1, \ldots, \sigma_n\}]) = Inst([Q(\vec{x})]) - Inst([Q(\vec{x})\sigma_1]) - \ldots - Inst([Q(\vec{x})\sigma_n])$

DEFINITION. — *A ψ-form with a main form that does not contain any variables and thus represents a single ground Kn or KW atom is called a* singleton*. When the main form contains at least one variable, we call it* non-ground*, or* quantified.

4. PSIPLAN-S

We remind the reader that PSIPLAN-S assumes infinite number of distinct domain objects.

DEFINITION (PSIPLAN-S PROPOSITION). — *A* PSIPLAN-S proposition *is either*

1) a Kn-atom of the form $Kn(a)$, where a is a ground atom, or
2) a Kn-ψ-form or KW-ψ-form.

We also use terms Kn-(or KW-)proposition to refer to any proposition that uses Kn (or KW).

DEFINITION (PSIPLAN-S STATE OF KNOWLEDGE - SOK). — PSIPLAN-S*'s State Of Knowledge is a satisfiable set of* PSIPLAN-S *Kn-atoms and Kn-ψ-forms that represent agent's knowledge about the world.*

DEFINITION (PSIPLAN-S GOAL). — *A* PSIPLAN-S *goal is any* PSIPLAN-S *proposition.*

KW-ψ-forms in PSIPLAN-S are used to represent information goals and results of sensing actions. They are used in reasoning about knowledge and ignorance. Consider $\tilde{\psi} = [KW(\neg PrefBib(x))]$, for example. Posted as a goal, $\tilde{\psi}$ requires knowing the value of each ground instance of $PrefBib(x)$, or in other words, knowing the set of preferred bibliographies. On the other hand, $\tilde{\psi}$ describes the effect of a sensing action that identifies the set of all preferred bibliographies.

A negated *KW*-proposition $\neg KW(p)$ represents ignorance about p. Although negated *KW*-propositions are not part of PSIPLAN-S, the $CKWA$ (see Definition on page 11) allows us to conclude that p is unknown, if $KW(p)$ cannot be deduced.

Note also that reasoning in PSIPLAN-S is done *without* expanding the ψ-forms into universal base, i.e. into the set of all of its ground instances.

4.1. PSIPLAN-S *action language*

PSIPLAN-S distinguishes two types of actions: *domain actions* that change the world (e.g., an action of downloading a paper from a url), and *sensing actions* that do not change the world but only return information about it (e.g., querying a bibliography).

Each domain action has a list of *preconditions*, $\mathcal{P}$, identifying the conditions necessary for executability of an action, and an encoding of the effects of the action, called the *assert list*, $\mathcal{A}$. An action precondition is a PSIPLAN-S goal, as defined above. We assume that an action is deterministic and can change the truth-value of only a *finite* number of atoms. Furthermore, the assert list refers to all and only those propositions that have changed their truth value as a result of action execution, thus each assert list is a set of propositions of the form $Kn(a)$ or $[Kn(\neg a)]$, where $a \in \mathcal{L}_{\mathcal{D}}$, i.e. is a ground atom. An action is *executable* in a world state w, if all action's preconditions are *true* in w.

Action *RmDir(?d)* of deleting an empty directory $?d$, depicted in Figure 2 is an example of a PSIPLAN-S domain action. Its preconditions are a *Kn*-atom and a quantified *Kn*-ψ-form. Its assert list consists of a *Kn*-ψ-form denoting a single *Kn*-atom.

Domain action $RmDir(?d)$
$\mathcal{P}: Kn(Dir(?d)), [Kn(\neg In(x, ?d))]$
$\mathcal{A}: [Kn(\neg Dir(?d))]$

Sensing action $QueryBib(?b, ?kwd)$
$\mathcal{P}: Kn(PrefBib(?b))$
$\mathcal{K}: [KW(\neg Rel(y, ?kwd) \vee \neg InCollection(y, ?b))]$

Figure 2. *Example of* PSIPLAN-S*'s domain and sensing actions. Variables x and y are implicitly universally quantified. Symbols starting with ? denote action schema parameters.*

Sensing actions also have preconditions. Effects of the sensing are specified by its *knowledge list*, denoted $\mathcal{K}$. The propositions in $\mathcal{K}$ are *KW*-ψ-forms. After a sensing action is executed, it returns an *observation list* of *Kn*-propositions corresponding to the information that was learned, denoted Δ. When the knowledge list $\mathcal{K}$ of a sensing action contains a quantified *KW*-ψ-form, i.e. denotes a statement about all domain objects, one of the propositions of Δ will also necessarily be a quantified ψ-form, since our assumption is that there is always a finite number of “positive” facts and representing the infinite number of “negative” facts requires quantification.

For example, *QueryBib(?b, ?kwd)*, depicted in Figure 2 is a sensing action that identifies all papers that according to bibliography $?b$ are related to keyword $?kwd$. The precondition requires that $?b$ be a preferred bibliography. The effect of this action is encoded in the knowledge list that contains a quantified ψ-form, and states that as a result of this action the set of all papers in collection of bibliography $?b$ that are related to keyword $?kwd$ in the title will be known.

Suppose after executing sensing action $a = QueryBib(RI, K)$ with effect $[KW(\neg Rel(p, K) \vee \neg InCollection(p, RI))]$ papers $Paper_1$ and $Paper_2$ were found as the only ones containing keyword K in their title, i.e. $\Delta(a)$ consists of the following propositions:

$$\begin{array}{l}[Kn(\neg Rel(p, K) \vee \neg InCollection(p, RI)) \text{ except } \{\{p = Paper_1\}, \{p = Paper_2\}\}] \\ Kn(Rel(Paper_1, K)), Kn(InCollection(Paper_1, RI)) \\ Kn(Rel(Paper_2, K)), Kn(InCollection(Paper_2, RI))\end{array} \quad (5)$$

4.2. *Reasoning in* PSIPLAN-S

In this section we consider entailment in PSIPLAN-S. We first address entailment of *Kn* and *KW* propositions (representing *domain* and *knowledge* goals) from a set of PSIPLAN-S *Kn*-propositions. This kind of reasoning is applied when sensing actions are not included in planning, as sensing actions have *KW*-propositions as their effects. We show that in PSIPLAN-S reasoning about domain and knowledge goals is sound and complete; we also provide a rationale for why its complexity is polynomial. This combination of soundness, completeness and tractability is unique among all implemented open-world planning formalisms.

Furthermore, we consider entailment from a set of *Kn* and *KW*-propositions, which is used when sensing actions are applied by the planner to satisfy knowledge goals. We present the sufficient conditions for entailment in such situations and illustrate its use in planning with examples.

4.3. *Reasoning without sensing actions*

First, consider entailment between two *simple* ψ-forms, i.e. ψ-forms without exceptions. In the following Lemma and below, whenever the $\subseteq$ sign is used to relate *Kn* or *KW* propositions, we actually mean the subset relationship between the domain literals within the parentheses that follow. For example, we would write $Kn(\neg P \vee \neg Q) \subseteq Kn(\neg R \vee \neg Q \vee \neg P)$, since $\{\neg P, \neg Q\} \subseteq \{\neg R, \neg Q, \neg P\}$.

LEMMA 8. — *Let ψ and ψ' denote simple Kn-ψ-forms , and $\tilde{\psi}'$ denote a simple KW-ψ-form.*

– *$\psi \models_k \psi'$ if and only if there exists a substitution σ such that $\mathcal{M}(\psi)\sigma \subseteq \mathcal{M}(\psi')$.*
– *$\psi \models_k \tilde{\psi}'$ if and only if $\psi \models_k \psi'$, where ψ' is obtained by replacing KW in $\tilde{\psi}'$ by Kn.*

When ψ is a singleton, the entailment is thus achieved in case of subsumption between the main forms of ψ and ψ'. In a general case, this Lemma reduces entailment to the existence of a substitution that matches the main form of the entailing *Kn-ψ*-form ψ onto a subset of clauses of the entailed ψ-form. In case such substitution exists, for each ground instance c of the entailed ψ-form there exists a ground instance in the entailing ψ-form ψ which *alone* entails c. Thus, the ground instances of ψ never need to be combined together to establish entailment. This property plays a critical role in the efficiency of ψ-form reasoning.

The property critical for the efficiency of reasoning is formulated in Theorem 9 below: given a set of *Kn-ψ*-forms $\Psi = \{\psi_1, \ldots, \psi_n\}$, $\Psi \models_k \psi$ only if there exists $\psi_i \in \Psi$ that *nearly entails* ψ, i.e. main part of ψ_i entails the main part of ψ, or $[\mathcal{M}(\psi_i)] \models_k [\mathcal{M}(\psi)]$. Note that this is a necessary condition only when the domain of objects is assumed to be infinite, which is the assumption that PSIPLAN-S makes.

THEOREM 9. — *Given a set of Kn-ψ-forms $\Psi = \{\psi_1, \ldots, \psi_n\}$ and a Kn-ψ-form ψ, $\Psi \models_k \psi$ only if there is $\psi_i \in \Psi$ such that $[\mathcal{M}(\psi_i)] \models_k [\mathcal{M}(\psi)]$.*

Theorem 9 thus requires that to satisfy a ψ-form goal, the planner must find an effect ψ_i that *nearly entails* ψ. But, as exceptions of ψ_i weaken it, ψ_i alone may not entail the entire ψ. To represent the propositions of ψ which are not entailed, we introduce the *e-difference*[3] operator.

DEFINITION. — *For any two sets of ground propositions A and B,* e-difference *is defined as follows:*

$$B \overset{e}{-} A = \{b \mid b \in B \ \wedge \ A \not\models_k b\}.$$

As ψ-forms are compact representations of sets of ground propositions, we extend the e-difference operation to ψ-forms. The algorithms for computing the e-difference between ψ-forms are not presented in this paper, however, we note that they are obtained in a straightforward manner from the corresponding Theorems (see (Babaian *et al.*, 2006)). The computation is carried out by manipulations on the main form and exceptions of the ψ-forms without expanding the ψ-form into the corresponding set of ground propositions. The following example illustrates the e-difference operation.

EXAMPLE 10. — Assume

$$\psi = [Kn(\neg PrefBib(z)) \text{ except } \{\{z = ACM\}, \{z = RI\}\}],$$

and let $\tilde{\psi} = [KW(\neg PrefBib(x) \vee \neg InCollection(y,x) \vee \neg Rel(y, Kwrd))]$, which can represent a goal of knowing for all papers y if they are related to keyword *Kwrd*, and are in collection of any preferred bibilography x.

ψ entails *most* of $\tilde{\psi}$, indeed, since $Kn(\neg PrefBib(x))$ is $true$ for all values of x except possibly RI and ACM, then so is the disjunction inside the $\tilde{\psi}$'s KW clause. Thus, the only parts of $\tilde{\psi}$ that are not entailed by ψ are

$$\tilde{\psi}_1 = [KW(\neg PrefBib(RI) \vee \neg InCollection(y, RI) \vee \neg Rel(y, Kwrd))]$$
$$\tilde{\psi}_2 = [KW(\neg PrefBib(ACM) \vee \neg InCollection(y, ACM) \vee \neg Rel(y, Kwrd))]$$

and therefore $\tilde{\psi} \overset{e}{-} \psi = \{\tilde{\psi}_1, \tilde{\psi}_2\}$.

This example demonstrates how e-difference is used in planning to identify the part of a non-ground goal $\tilde{\psi}$ that is not implied by the existing knowledge ψ. Indeed, knowing that there are no preferred bibliographies except possibly RI and ACM means no bibliographies except for possibly these two need to be considered, thus reducing the knowledge goal $\tilde{\psi}$ to two simpler goals $\tilde{\psi}_1$ and $\tilde{\psi}_2$. □

The e-difference operator also plays a key role in computing entailment. The next Theorem describes the necessary and sufficient conditions for entailment of a *Kn*, or *KW* ψ-form by a set of atoms and *Kn*-ψ-forms.

DEFINITION. — *Set s of Kn and KW-propositions is called* saturated *if for every $Kn(a)$ and $Kn(\neg a \vee \neg a_1 \vee \ldots \vee \neg a_n)$ in s the resolvent $Kn(\neg a_1 \vee \ldots \vee \neg a_n)$ is also in s.*

3. 'e' in e-difference refers to *entailment* relation, used in the definition.

In other words, s is saturated when the application of rule (R1) does not generate any *new* propositions not already in s. A saturated equivalent of a set can always be computed by performing all possible such resolutions.

Although computing a saturated equivalent of a set is a relatively computationally expensive operation (see (Babaian *et al.*, 2006) for the algorithm and its analysis), planning with PSIPLAN-S requires that it is performed only once on the initial state and, in a more bounded form, upon executing a sensing action. Execution of a sensing action adds new Kn-atoms and ψ-forms to the the SOK; in such cases the saturation computation is limited to the newly added propositions.

THEOREM 11. — *Let* $s = A \cup \Psi$ *be a satisfiable saturated set of* PSIPLAN-S *Kn-atoms (A) and Kn-*ψ*-forms (*Ψ*),* $s \models_k \psi$ *if and only if*

1) $\psi = [KW(\neg a_1 \vee \ldots \vee \neg a_n)]$ *and* $Kn(a_1), \ldots, Kn(a_n) \in A$*, or*
2) there exists $\psi_k \in \Psi$*, such that* $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\psi)]$*, and, furthermore,* $s - \{\psi_k\} \models_k (\psi \overset{e}{-} \psi_k)$

Clause 1 of the above Theorem specifies that one way of establishing entailment of a singleton *KW-*ψ-form (i.e. a ψ-form without any variables, i.e. denoting a single ground proposition) is to find a set of corresponding *Kn*-atoms.

Clause 2 applies to any (singleton or quantified) ψ-form and requires finding a near entailing ψ-form ψ_k, followed by checking entailment of the part of ψ that is not entailed by ψ_k. Thus, the Theorem describes a recursive procedure for checking entailment. Importantly, each ψ-form in the result of e-difference $\psi \overset{e}{-} \psi_k$ will have fewer variables than there are in ψ. This factor limits the depth of the recursion tree to the maximum number of variables in a ψ-form.

Overall, the complexity of the entire procedure of checking entailment $s \models_k \psi$ is $O(\beta^V n)$, where n is the number of *Kn*-atoms and ψ-forms in s, V is the maximum number of variables in a ψ-form, and β is the maximum number of ψ-forms in the e-difference. We assume that the cardinality of predicate symbols is constant bounded, and unification takes constant bounded time. We also assume the number of variables, literals and exceptions are all constant bounded. In that case β is also constant bounded, and checking entailment has a linear complexity bound. When the maximum number of exceptions is proportional to n, the procedure's complexity is bound by a polynomial of the degree $V + 1$.

4.4. *Adding sensing actions*

When sensing actions are added, the planner must reason from a set of propositions which include both *Kn* and *KW* ψ-forms. Since the *KW*-propositions do not contribute to the entailment of *Kn*-propositions, we consider entailment of only knowledge goals, i.e. *KW-*ψ-forms.

First, we present an analog of Lemma 8, establishing the necessary and sufficient conditions for entailment between two simple *KW*-ψ-forms. However, to achieve an important property of *Kn*-ψ-form entailment which follows from Lemma 8 in the case of *KW*-ψ-form entailment, namely, the fact that each ground proposition of the entailed ψ-form is always entailed by some ground proposition of the entailing ψ-form *alone*, we must require that the main form of each *KW*-ψ-form does not contain duplicate predicate symbols. Consider the following example:

EXAMPLE 12. — Let

$$\begin{aligned} \tilde{\psi}_1 = & \quad [KW(\neg P(x,y))] \\ \tilde{\psi}_2 = & \quad [KW(\neg P(A,B) \vee \neg P(C,D))]. \end{aligned}$$

No single ground instance of of $\tilde{\psi}_1$ alone would imply that the value of the disjunction $\neg P(A,B) \vee \neg P(C,D)$ is known. However, taking two ground instances $KW(\neg P(A,B))$ and $KW(\neg P(C,D))$ of $\tilde{\psi}_1$ together allows to conclude that the value of the disjunction $\neg P(A,B) \vee \neg P(C,D)$ is known, thus together they entail $KW(\neg P(A,B) \vee \neg P(C,D))$.

□

LEMMA 13. — *Let $\tilde{\psi}$, $\tilde{\psi}'$ denote simple KW-ψ-forms without repeated predicate symbols in the main form. $\tilde{\psi} \models_k \tilde{\psi}'$ if and only if there exists a substitution σ such that $\mathcal{M}(\tilde{\psi})\sigma = \mathcal{M}(\tilde{\psi}')$.*

The added ways of proving a *KW*-ψ-form goal are based on inference rules (R5) and (R6). We precede the formulation of the corresponding Theorem with two examples that illustrate it.

EXAMPLE 14. — Rule (R5) essentially states that the value of a disjunction (or conjunction) can be determined from the values of its subclauses. To illustrate the idea, consider the ψ-form goal from the previous example:

$$\tilde{\psi}_1 = [KW(\neg PrefBib(RI) \vee \neg InCollection(y, RI) \vee \neg Rel(y, Kwrd))].$$

The above goal can be achieved by a combination of sensing operators with effects of knowing wether RI is a preferred bibliography ($[KW(\neg PrefBib(RI))]$), and knowing all papers in collection of the ResearchIndex that are related to keyword $Kwrd$, which is expressed with a ψ-form $[KW(\neg InCollection(y, RI) \vee \neg Rel(y, Kwrd))]$.

□

Another example illustrates application of rule $(R6)$. This rule is based on the fact that the value of a clause c can be determined by resolution between an atom a and the clause $\neg a \vee c$.

EXAMPLE 15. — Imagine a goal of finding out if paper P was published in journal J, i.e. $[KW(\neg Published(J, P))]$. Suppose there was a sensing action $PublishedPapersByAuthor(J, A)$ with the effect $[KW(\neg Published(J, x) \vee \neg Author(x, A))]$ identifying all papers by author A published by journal J. If it is known that A is indeed an author of P, i.e. $Kn(Author(P, A))$, then after executing the $PublishedPapersByAuthor(J, A)$ action it will be known whether P was published in J, in other words, the goal $[KW(\neg Published(J, P))]$ will be achieved.

□

Differently from the reasoning without sensing, we do not make the completeness case for reasoning about *KW*-propositions from a set of *Kn* and *KW*-propositions, because we have not shown completeness of the ground language $\mathcal{L}_{\mathcal{K}}$ with respect to this kind of reasoning. However, we believe it to be true, based on the Conjecture on page 11.

THEOREM 16. — *Let A denote a set of Kn-atoms $\{Kn(a_1), \ldots, Kn(a_k)\}$, Ψ denote a set of Kn-ψ-forms $\{\psi_1, \ldots, \psi_r\}$ and $\tilde{\Psi}$ denote a set of KW-ψ-forms $\{\tilde{\psi_1}, \ldots, \tilde{\psi_t}\}$. Each KW-$\psi$-form of $\tilde{\Psi}$ does not have repeated predicate symbols in its main form.*

Assume $A \cup \Psi \cup \tilde{\Psi}$ is satisfiable and saturated. Let $\tilde{\psi}$ be a KW-ψ-form without repeated predicate symbols in the main form. Then, $A \cup \Psi \cup \tilde{\Psi} \models_k \tilde{\psi}$ if at least one of the following conditions holds:

1) There are $Kn(a_1), \ldots, Kn(a_n) \in A$, such that $\tilde{\psi} = [KW(\neg a_1 \vee \ldots \vee \neg a_n)]$.

2) There is a $\psi_k \in \Psi \cup \tilde{\Psi}$ such that $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\tilde{\psi})]$, and $A \cup \Psi \cup \tilde{\Psi} - \psi_k \models_k \tilde{\psi} \overset{e}{-} \psi_k$.

3) There is a decomposition of the main part of $\tilde{\psi}$ into two subclauses Q' and Q'' such that $\mathcal{M}(\tilde{\psi}) = Q' \vee Q''$ and

$$A \cup \tilde{\Psi} \models_k [KW(Q') \,|\, \Sigma'], \quad \textit{and}$$
$$A \cup \tilde{\Psi} \models_k [KW(Q'') \,|\, \Sigma''],$$

where Σ', Σ'' are subsets of the set of exceptions of $\tilde{\psi}$, which only use variables from Q' and Q'', respectively.

4) Assume $\tilde{\psi} = [KW(Q)$ except $\Sigma]$. There is an atom $Kn(a_i) \in A$ and $\tilde{\psi}' \in \tilde{\Psi}$ such that $\tilde{\psi}' \models_k [KW(\neg a_i \vee Q)$ except $\Sigma]$.

Cases 1 and 2 of this Theorem repeat the conditions of Theorem 11. Cases 3 and 4 describe entailment that involves *KW*-premises. These entailment methods were demonstrated in Examples 14 and 15.

Unlike the case of only *Kn*-premises, the completeness of these entailment checking methods is not guaranteed, and their complexity is outside of the polynomial bounds in the number of propositions, since different combinations of *KW*-ψ-forms and *Kn*-atoms must be considered to cover cases 3 and 4. In our experience with using the language for planning in the Writer's Aid application, however, this theoretically exponential upper bound did not present a practical problem, due to a small number of cases that required application of cases 3 and 4 of the Theorem.

4.5. *SOK update after an action and projection*

Domain actions cause transitions between the world states. As actions are executed, the agent's SOK must evolve in parallel with the world, and must adequately reflect the changes that occur due to an action. Sensing actions have no effect on the world state, however, the SOK should also be updated with the facts learned upon execution of a sensing action. For the purpose of planning, the agent should have the ability to correctly predict what its state of knowledge would be in case a particular sequence of steps is executed. In this section we present the appropriate PSIPLAN-S operations and discuss their properties.

SOK update. We remind the reader that PSIPLAN-S assumes that action effects are deterministic and that the description of each action is complete and correct, i.e. there are no unknown effects or preconditions.

DEFINITION. — *Let w denote a world state and a denote an action that is executable in w. By $do(a, w)$ we denote the world that results from executing a in w. For a set of world states W we define $do(a, W)$ - a set of worlds resulting from performing action a in any of the world states in W, assuming a is executable in any of the worlds in W, as follows:*

$$do(a, W) = \begin{cases} W, \text{ \textit{if a is a sensing action,}} \\ \{w' | w' = do(a, w), w \in W\}, \text{ \textit{if a is a domain action .}} \end{cases}$$

Let $update(s, a)$ denote the SOK transition that is caused by an agent with SOK s taking action a. *Correctness* of an SOK update guarantees that the SOK always correctly represents the actual world model, given a correct initial SOK. The other desirable property of the SOK update is *completeness*: we would like the agent to maintain all the valid knowledge and take into account the information observed via sensing. The correctness and completeness properties of the SOK update are necessary for a sound and complete planning algorithm.

The correctness and completeness criteria can be formulated in the context of possible worlds. We remind the reader that $\Delta(a)$ denotes the set of PSIPLAN-S propositions that are *observed* as opposed to *caused* during the execution of an action, thus $\Delta(a)$ for a domain action denotes an empty set.

DEFINITION. — *We say that the update procedure is* correct *if and only if*

$$\mathcal{B}(update(s, a)) \subseteq do(a, \mathcal{B}(s \cup \Delta(a))) \tag{6}$$

i.e. every possible world after performing the action a has to have a possible predecessor among the worlds in which the propositions that were observed via sensing are true.

The update procedure is complete *if and only if*

$$do(a, \mathcal{B}(s \cup \Delta(a))) \subseteq \mathcal{B}(update(s, a)) \tag{7}$$

i.e. every world obtained from a previously possible world in which the propositions that were observed via sensing are true*, is accessible from the new SOK.*

To achieve correctness of SOK updates, the agent must remove from the SOK all propositions whose truth value might have changed as the result of the performed action. In order to be complete, the agent must also add to the SOK all facts that become known.

To update SOK s after performing a domain action a_d all propositions whose truth value could have been changed must be removed from s – these are all propositions entailed by the *negation* of some effect of a_d. The propositions entailed by effects of a_d are also removed, and then the effects of a_d are added to the new SOK. The agent's SOK after executing a domain action a_d in the SOK s is computed as shown below.

$$update(s, a_d) = ((s \overset{e}{-} \mathcal{A}^-(a_d)) \overset{e}{-} \mathcal{A}(a_d)) \cup \mathcal{A}(a_d), \tag{8}$$

where $\mathcal{A}^-(a_d)$ denotes the set of propositions obtained by negating each domain proposition inside the $Kn()$ in a_d's assert list $\mathcal{A}(a_d)$.

EXAMPLE 17. — For an example, consider action $a = Download(P, U, D)$ of downloading file P from url U into directory D. Let precondition $\mathcal{P}(a) = \{Kn(Location(P, U))\}$ denote that P is located at url U. The effect of this action is that file P is stored in directory D, i.e. $\mathcal{A}(a) = \{Kn(In(P, D))\}$.

We begin with an SOK s, which states that location of paper P is U, and that A is the only Postscript file in the system. $T(x, PS)$ below denotes that the type of file x is Postscript.

$$s = \left\{ \begin{array}{l} Kn(Location(P, U)), Kn(T(A, PS)), \\ [Kn(\neg In(x, y) \vee \neg T(x, PS)) \text{ except } \{\{x = A\}\}] \end{array} \right\}$$

$a = Download(P, U, D)$ is executable in s, and $\mathcal{A}^-(a) = \{[Kn(\neg In(P, /img))]\}$. The e-difference $s \overset{e}{-} \mathcal{A}^-(a)$ results in the exception $\{x = P, y = D\}$ being added to the ψ-form. Further e-difference with $\mathcal{A}(a)$ and union with $\mathcal{A}(a)$ yields the following SOK

$$s' = \left\{ \begin{array}{l} Kn(Location(P, U)), Kn(T(A, PS)), \\ [Kn(\neg In(x, y) \vee \neg T(x, PS)) \text{ except } \{\{x = A\}, \{x = P, y = D\}\}] \\ Kn(In(P, D)) \end{array} \right\}$$

Note that s entailed $Kn(\neg In(P, D) \vee \neg T(P, PS))$ and that we added the effect $Kn(In(P, D))$ when determining s'. If our update rule retained $Kn(\neg In(P, D) \vee \neg T(P, PS))$ as a part of a ψ-form in s', then in s' we could perform resolution and conclude that $\neg T(P, PS)$. However, this would be wrong because we have no information on whether or not P is a Postscript file. Instead, our update rule removes any clause that is entailed by $\neg In(P, D)$, and so s' does not entail $\neg In(P, D) \vee \neg T(P, PS)$.

□

The update (8) produces the same result as Winslett's update operator (Winslett, 1988) in the special case where actions are deterministic. The update is computed without considering all possible worlds corresponding to SOK s explicitly, and thus is efficient.

After the execution of a sensing action a_s, the set of observed propositions, denoted below by $\Delta(a_s)$ is added to the SOK, i.e.

$$update(s, a_s) = s \cup \Delta(a_s) \quad (9)$$

After propositions from $\Delta(a)$ are added to the SOK, all possible resolutions from SOK propositions must be computed and added to the new SOK to keep it saturated.

THEOREM 18. — *Assume s is satisfiable and saturated. The following state of knowledge update procedure is correct and complete:*

$$update(s, a) = \begin{cases} ((s \overset{e}{-} \mathcal{A}^{-}(a)) \overset{e}{-} \mathcal{A}(a)) \cup \mathcal{A}(a), & \text{when } a \text{ is a domain action,} \\ s \cup \Delta(a), & \text{when } a \text{ is a sensing action.} \end{cases}$$

SOK projection. In order to plan, it is necessary to predict the state of knowledge resulting from executing an action. This is realized using the projection operation, $project(s, a)$. For a domain action, assuming no execution failure, $project(s, a_d) = update(s, a_d)$.

For a sensing action projection is different from updating, since it occurs prior to executing the sensing action and no results of sensing are available:

$$project(a_s, s) = s \cup \mathcal{K}(a_s). \quad (10)$$

Projection of a sensing action, thus, simply adds the knowledge list to the agent's SOK, establishing that in the future SOK the value of some propositions will become known, without committing to the actual truth value of those propositions. Note that the result of $project(a_s, s)$ is not a SOK, since PSIPLAN-S SOKs do not contain *KW*-propositions. We call the result of projection a *projected* SOK, or $PSOK$, denoted $\tilde{s}$, and apply the Closed Knows Whether Assumption to PSOKs as we do with SOKs: $CKWA(\tilde{s}) = \tilde{s} \cup \{\neg KW(p) \,|\, p \in \mathcal{L_D} \wedge \tilde{s} \not\models_k KW(p)\}$.

Note that the result of the projection does not change the set of possible worlds corresponding to the SOK, i.e.

$$\mathcal{B}(project(a_s, s)) = \mathcal{B}(s),$$

but it has fewer models, since those models of s that are uncertain about the value of propositions in $\mathcal{K}(a_s)$ are not part of the PSOK $project(a_s, s)$.

EXAMPLE 19. — Using the projection operator and Proposition 3, we can show that the goal

$$\tilde{\psi}_1 = [KW(\neg PrefBib(RI) \vee \neg InCollection(y, RI) \vee \neg Rel(y, Kwrd))]$$

can be established by a combination of two sensing actions: $a_1 = CheckPreferred(RI)$ with effect $[KW(\neg PrefBib(RI))]$, i.e. finds out if RI is a preferred bibliography, and $a_2 = QueryBib(RI, Kwrd)$ with effect $[KW(\neg InCollection(y, RI) \vee \neg Rel(y, Kwrd))]$, since $project(\{a_1, a_2\}, s) \models_k \tilde{\psi}_1$ (the entailment follows from case 3 of Theorem 16).

Note further that if a_1 is executed first and returns with observation $[Kn(\neg PrefBib(RI))]$, this proposition would be added to the SOK by update (9), and the updated SOK will entail $\tilde{\psi}_1$ (as follows from case 2 of Theorem 16), rendering this goal satisfied.

□

5. Related work

Scherl and Levesque (Scherl *et al.*, 1993; Scherl *et al.*, 2003) model the agent's knowledge as an accessibility relation K over the situations (corresponding to *worlds* in our terminology) and define the set of possible worlds as a set of all K-accessible situations. Scherl and Levesque's formulation of knowing P is expressed in terms of the K relation as follows $Know(P, w) = \forall w'.K(w, w') \Rightarrow w'(P)$. Thus, the changes in the agent's beliefs are expressed as changes to the K relation in a form of successor-state axioms for the K relation. The successor-state axiom for the K relation is shown to have the following properties (Scherl *et al.*, 2003):

1) The knowledge producing actions do not change the state of the world.

2) Knowledge is updated appropriately to the effects of an action: only valid knowledge is added (default persistence of ignorance), no valid knowledge is discarded (memory) and all knowledge that becomes available as a result of a sensing action is added.

In our framework the set of possible worlds is modeled using the set $\mathcal{B}(s)$, where s is the explicit and finite representation of the agent's knowledge state. Knowing P is defined as $s \models_k P$, because in such case it follows that $\forall w \in \mathcal{B}(s).w(P)$. Knowledge updates and projections in PSIPLAN-S are made via application of the *update* and *project* operators defined in 8-10. The update and projection operations share the properties of knowledge update of Scherl and Levesque listed in the previous paragraph.

A subset of PSIPLAN-S that does not include KW-propositions and sensing actions is presented in (Babaian *et al.*, 2006). PSIPLAN-S's use of the modal operators *Kn* and *KW* is inspired by the theoretical foundation for representing knowledge and sensing and the solution to the frame problem developed by Scherl and Levesque (Scherl *et al.*, 1993). Their solution is developed within the framework of the *situation calculus* (Reiter, 1991) in which general reasoning is not decidable. The goal of PSIPLAN-S is to provide a representation for open world planning with sensing that uses the modal operators *Kn* and *KW* (Scherl *et al.*, 1993; Scherl *et al.*, 2003) in a less expressive language for the sake of decidability and tractability.

Liu and Levesque (Liu *et al.*, 2005) present a subset of situation calculus (McCarthy *et al.*, 1969; Reiter, 2001) with equality with complete and tractable reasoning. The completeness of determining whether a formula ϕ is true after executing an action is achieved by restricting ϕ to be in a certain normal form, the knowledge base to a certain *proper* form and restricting the actions to those with quantifier-free conditional effects. Furthermore, the preconditions of each conditional effect must be known in the knowledge base which describes the situation in which the action is applied. The language of Liu and Levesque does not subsume PSIPLAN-S, however there is some overlap between the two. In particular, while ψ-forms with a single predicate in the main form can be represented in their language, it does not represent ψ-forms in which the main form is a disjunction, e.g. $[Kn(\neg InCollection(y, ACM) \vee \neg Rel(y, Kwrd))]$. An example of a statement from their language that cannot be expressed in PSIPLAN-S is a universally quantified positive statement, e.g.: $\forall x.P(x) \vee x = A$. However, statements like that do not seem to come up in planning in domains with infinite number of objects.

FLUX (Thielscher, 2005) is a framework for programming agent behavior for environments with incomplete information and sensing. Agent programming is different from the problem of automated planning, which is where PSIPLAN-S has been used. FLUX is based on fluent calculus and is implemented as a set of constraints, defining the domain, action update, agent's knowledge and action execution. The constraint language allows linear time evaluation, and includes universally quantified negated clauses, similar to the simple ψ-forms of PSIPLAN-S. However, the domain of objects is assumed to be finite.

ASCP planner (Tu *et al.*, 2007) is based on an extension of the 0-approximation of sensing actions in action language $\mathcal{A}$ by (Son *et al.*, 2001). ASCP encodes the planning problem as a logic program, with the solution plan derived from the logic program's answer set. $\mathcal{A}$ includes conditional effects, which are excluded in PSIPLAN-S. While conditional planning within the 0-approximation is shown to be NP-complete, 0-approximation sacrifices the completeness of conditional planning (see also (Babaian *et al.*, 2006), page 36).

Work by Petrick and Bacchus (Petrick *et al.*, 2002; Petrick *et al.*, 2004) is closely related to PSIPLAN-S's representation. Their knowledge state representation uses propositions similar to PSIPLAN-S's *Kn*, and *KW*-propositions, and their sensing actions have *KW*-effects. While they use quantified propositions like our $[KW(Q(x))]$, but without exceptions, to represent the fact that the agent knows the truth values of all ground instances of $[KW(Q(x))]$, their goals are limited to ground propositions. Their representation is based on the LCW-representation (Etzioni *et al.*, 1997), which is tractable but incomplete. The LCW representation has also been used in PUCCINI planner (Golden, 1998). By contrast, in PSIPLAN-S reasoning about domain and knowledge goals from an agent's SOK is complete and tractable, and reasoning from a projected SOK is tractable and, we conjecture, complete.

The action languages in both PUCCINI and planner of Bacchus and Petrick are different from PSIPLAN-S's: there are no universally quantified preconditions, but there are conditional effects of actions. Conditional effects specify propositions that become true if a certain condition holds prior to taking the action. While conditional effects can be easily incorporated in PSIPLAN-S, by excluding conditional effects from PSIPLAN-S actions, we are able to specify exactly the set of propositions whose truth value is changed when executing an action, thus contributing to the tractability of reasoning without having to sacrifice completeness. Planning with conditional effects and without sensing has been shown (e.g. (Turner, 2002; Baral *et al.*, 2000)) to be outside of the NP class of problems. Such limitations on action specification does not seem to preclude PSIPLAN-S's applicability to some real-life domains, as demonstrated by its use in Writer's Aid (Babaian *et al.*, 2002).

6. Conclusions and future work

We have presented PSIPLAN-S, a representation language for reasoning and planning in open world applications with sensing. Several distinguishing features are critical for PSIPLAN-S's suitability in practice.

– PSIPLAN-S is based on a language of knowledge propositions $\mathcal{L}_k$, which was carefully designed to support sound and complete reasoning about domain and knowledge goals from the agent's state of knowledge.

– PSIPLAN-S introduces a novel Closed Knows Whether Assumption to represent the agent's ignorance. Verifying an agent's ignorance about a proposition is a necessary component of ensuring non-redundant sensing.

– To represent statements that are quantified over the entire domain of discourse, PSIPLAN-S uses ψ-forms, which represent infinite sets of propositions of $\mathcal{L}_\mathcal{K}$. Completeness of reasoning from an agent's SOK in PSIPLAN-S follows from the same properties of reasoning in $\mathcal{L}_\mathcal{K}$ and ψ-forms.

– PSIPLAN-S's SOK update after an action completely and correctly reflects the transition due to an action without expanding the SOK into the set of possible worlds. Although we have not shown it here, the update procedure has polynomial complexity in the number of propositions in SOK (Babaian *et al.*, 2006).

– PSIPLAN-S's projection operator predicts the future SOK and serves as a basis for planning with sensing actions.

Thus, PSIPLAN-S efficiently handles domains with an incomplete specification of the initial state without considering the set of all possible worlds, and does not require that the agent know the set of all objects in the domain. In the future, we will extend PSIPLAN-S to allow function symbols. Another interesting and important task is proving the completeness of the rule system for $\mathcal{L}_\mathcal{K}$.

Acknowledgements

We thank the anonymous reviewers for their insightful and constructive comments that helped improve this paper.

7. References

Babaian T., Grosz B. J., Shieber S. M., “A Writer’s Collaborative Assistant”, *"Proceedings of Intelligent User Interfaces’02"*, ACM Press, pp. 7-14, 2002.

Babaian T., Schmolze J., “PSIPLAN: open world planning with ψ-forms”, *Artificial Intelligence Planning and Scheduling: Proceedings of the Fifth International Conference (AIPS’00)*, pp. 292-300, 2000.

Babaian T., Schmolze J. G., “Efficient Open World Reasoning for Planning”, *Logical Methods in Computer Science*, vol. 2, num. 3, pp. 1-39, 2006.

Baral C., Kreinovich V., Trejo R., “Computational complexity of planning and approximate planning in the presence of incompleteness”, *Artificial Intelligence*, vol. 122, num. 1-2, pp. 241-267, 2000.

Carlin A., Schmolze J. G., Babaian T., “Graphplan Based Conformant Planning with Limited Quantification”, *Research on Computing Science*, vol. 16, Advances in Artificial Intelligence Theory, pp. 65-75, 2005.

Eiter T., Faber W., Leone N., Pfeifer G., Polleres A., “A logic programming approach to knowledge-state planning: Semantics and complexity”, *ACM Trans. Comput. Logic*, vol. 5, num. 2, pp. 206–263, 2004.

Etzioni O., Golden K., Weld D., “Sound and efficient closed-world reasoning for planning”, *Artificial Intelligence*, vol. 89, num. 1–2, pp. 113-148, January, 1997.

Golden K., “Leap Before You Look: Information Gathering in the PUCCINI planner”, *Artificial Intelligence Planning Systems: Proceedings of the Fourth International Conference (AIPS’98)*, AAAI Press, Carnegie Mellon University, Pittsburgh, Pennsylvania, June, 1998.

Levesque H. J., Reiter R., Lesperance Y., Lin F., Scherl R. B., “GOLOG: A Logic Programming Language for Dynamic Domains”, *Journal of Logic Programming*, vol. 31, num. 1-3, pp. 59-83, 1997.

Liu Y., Levesque H. J., “Tractable reasoning with incomplete first-order knowledge in dynamic systems with context-dependent actions”, *IJCAI*, pp. 639–644, 2005.

McCarthy J., Hayes P. J., “Some Philosophical Problems from the Standpoint of Artificial Intelligence”, *in* B. Meltzer, D. Michie (eds), *Machine intelligence 4*, American Elsevier, New York, 1969.

Moore R. C., “A formal theory of knowledge and action”, *in* J. R. Hobbs, R. C. Moore (eds), *Formal Theories of the Commonsense World*, Ablex, Norwood, New Jersey, pp. 319–358, 1985.

Petrick R., Bacchus F., “A Knowledge-Based Approach to Planning with Incomplete Information and Sensing”, *AI Planning and Scheduling (AIPS2002)*, pp. 212-222, 2002.

Petrick R. P. A., Bacchus F., “Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing”, *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-2004)*, AAAI Press, Menlo Park, CA, pp. 613–622, Jun, 2004.

Reiter R., “The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression”, *in* V. Lifschitz (ed.), *Artificial Intelligence and*

Mathematical Theory of Computation: Papers in Honor of John McCarthy, Academic Press, New York, pp. 359–380, 1991.

Reiter R., *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*, MIT Press, Bradford Books, Cambridge, MA, 2001.

Scherl R. B., Levesque H. J., "The Frame Problem and Knowledge-Producing Actions", *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, 1993.

Scherl R. B., Levesque H. J., "Knowledge, Action and the Frame Problem", *Artificial Intelligence*, vol. 144, num. 1-2, pp. 1-39, 2003.

Son T. C., Baral C., "Formalizing sensing actions – a transition function based approach.", *Artificial Intelligence*, vol. 125, num. 1-2, pp. 19-91, 2001.

Thielscher M., "FLUX: A logic programming method for reasoning agents", *Theory and Practice of Logic Programming*, vol. 5, num. 4-5, pp. 533-565, July, 2005.

Tu P. H., Son T. C., Baral C., "Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming", *Theory and Practice of Logic Programming*, vol. 7, num. 4, pp. 377–450, 2007.

Turner H., "Polynomial-length planning spans the polynomial hierarchy", *Proceedings of Eighth European Conf. on Logics in Artificial Intelligence (JELIA'02)*, 2002.

Winslett M., "Reasoning about action using a possible models approach", *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, 1988.

8. Proofs

We restate all propositions here for convenience.

PROPOSITION 1.— Let $p_1, \ldots, p_n, q$ be domain propositions:

$$(p_1, \ldots, p_n \models q) \;\Leftrightarrow\; (Kn(p_1) \ldots Kn(p_n) \models_k Kn(q)).$$

PROOF. — ($\Rightarrow$) First, suppose the set of propositions $\{(p_1, \ldots, p_n\}$ is unsatisfiable, i.e. has an empty set of models. Then, according to Definitions (2.3)-(2.3), so is the set $\{Kn(p_1) \ldots Kn(p_n)\}$, so the statement is true.

When $\{(p_1, \ldots, p_n\}$ is a satisfiable set, $p_1, \ldots, p_n \models q$ means that for every world w such that $w(p_1), \ldots, w(p_n)$ we have $w(q)$. Let (w, W) be a model for $Kn(p_1) \ldots Kn(p_n)$. Then, according to Definitions (2.3)-(2.3), for all worlds w in W, we have $w(p_1), \ldots, w(p_n)$. Hence for all w in W, we have $w(q)$. Therefore (w, W) is a model of $Kn(q)$.

($\Leftarrow$) First, suppose the set of propositions $\{Kn(p_1) \ldots Kn(p_n)\}$ is unsatisfiable. Then, there is no world w such that $w(p_1), \ldots, w(p_n)$, because otherwise $(w, \{w\})$ would be a model for $Kn(p_1) \ldots Kn(p_n)$. Therefore, $\{p_1, \ldots, p_n\}$ is also unsatisfiable and the statement holds true.

Now consider the case where $\{Kn(p_1) \dots Kn(p_n)\}$ is a satisfiable set. Suppose that $Kn(p_1) \dots Kn(p_n) \models_k Kn(q)$, but $p_1, \dots, p_n \not\models q$. The latter means that there is a world w such that $w(p_1), \dots, w(p_n)$, but not $w(q)$. Then, a k-state $(w, \{w\})$ is a model for $Kn(p_1) \dots Kn(p_n)$, but not for $Kn(q)$. We arrive at a contradiction.

■

PROPOSITION 2.— Let c be a domain clause.

1) If a is a domain atom, $Kn(a) \models_k KW(c)$ if and only if $c = \neg a$.
2) If c' is a domain clause $Kn(c') \models_k KW(c)$ if and only if $c' \subseteq c$.
3) If $a_1, \dots, a_r$ are domain atoms, and $c_1, \dots, c_n$ are domain clauses, and the set $\{a_1, \dots, a_r, c_1, \dots, c_n\}$ is satisfiable,

$$s = \{Kn(a_1), \dots, Kn(a_r), Kn(c_1), \dots, Kn(c_n)\} \models_k KW(c)$$

if and only if
a) there exists a set of domain atoms $a_{i_1}, \dots, a_{i_m}$, such that $Kn(a_{i_j}) \in s$ and $c = \neg a_{i_1} \vee \dots \vee \neg a_{i_m}$, or
b) there exists a domain clause c', such that $s \models_k Kn(c')$ and $c' \subseteq c$.

PROOF. — The *if* parts of claims 1-3 of this proposition are easily verified using the definitions of the set of models $\alpha()$ and entailment in Section 2.3. It is only left to prove the *only if* part. In all cases the proof is by contradiction. We show that if the assertion of each case of the theorem does not hold, there exists a model of the premise, that is not a model of the conclusion.

1. Suppose $Kn(a) \models_k KW(c)$, but $c \neq \neg a$. Assume $c = \neg b_1 \vee \dots \vee \neg b_r$, where $r \geq 1$ and $a \neq b_i$ for at least one value of i, where $1 \leq i \leq r$. Let γ_0 be a k-state $(w', \{w', w''\})$ such that $w'(a)$, $w''(a)$, $w'(b_i)$ and $w''(\neg b_i)$ for all $1 \leq i \leq r$ such that $a \neq b_i$. $\gamma_0 \in \alpha(Kn(a))$ and $\gamma_0 \notin \alpha(KW(c))$, which implies that $Kn(a) \not\models_k KW(c)$. We arrived at a contradiction.

2. Suppose $Kn(c') \models_k KW(c)$, but $c' \not\subseteq c$. The latter implies there exists a disjunct $\neg a$ of c', which is not a disjunct of c. Assume $c = \neg b_1 \vee \dots \vee \neg b_r$, where $r \geq 1$. Let γ_0 be a k-state $(w', \{w', w''\})$ such that $w'(\neg a)$, $w''(\neg a)$, $w'(b_i)$ for all $1 \leq i \leq r$, and $w''(\neg b_i)$ for all $1 \leq i \leq r$. Clearly, $\gamma_0 \in \alpha(Kn(c'))$ and $\gamma_0 \notin \alpha(KW(c))$, which implies that $Kn(c') \not\models_k KW(c)$. We arrived at a contradiction.

3. Suppose $s = \{Kn(a_1), \dots, Kn(a_r), Kn(c_1), \dots, Kn(c_n)\} \models_k KW(c)$, where the set $\{a_1, \dots, a_r, c_1, \dots, c_n\}$ is satisfiable, $c \neq \neg a_i$, for all $1 \leq i \leq r$, and there is no $c' \subseteq c$ such that $s \models_k Kn(c')$.

We first address the case when $n = 0$, i.e. there are no $Kn()$ formulas over clauses of negated atoms in s. Then, item (3.b) in the formulation of the proposition is not applicable, because, based on Proposition 1, there would be no clauses of negated atoms c' such that $s \models_k Kn(c')$. Assuming the assertion (3.a) of this Proposition does not hold, c is is not composed entirely of negations of atoms $a_1, \dots, a_r$, and therefore the set of literals $\neg b_p$ in c that are not one of $\neg a_1, \dots, \neg a_r$ is not empty. Let

$b_1, \ldots, b_p$ denote all literals of c that are not $a_1, \ldots, a_r$. Let's now construct a k-state $\gamma_0 = (w', \{w', w''\})$ where $w'(a_i), w''(a_i)$, for all $1 \leq i \leq r$, and set $w'(b_i)$ and $w''(\neg b_i)$ for all $1 \leq i \leq p$. It's easy to see that γ_0 is a model of s, but not of $KW(c)$.

In case $n > 0$, let $c'_1, \ldots c'_v$ be the set of *all* clauses that can be obtained from $\{a_1, \ldots, a_r, c_1, \ldots, c_n\}$ by unit clause resolution rule, plus those clauses from $C = \{c_1, \ldots, c_n\}$ that do not contain negations of any atoms from $A = \{a_1, \ldots, a_r\}$, i.e. do not produce a resolution with any atom from A. According to Proposition 1 and the fact that deduction from a satisfiable set in $\mathcal{L}_\mathcal{D}$ is complete under unit clause resolution and subsumption, for any c' such that $s \models_k Kn(c')$, there must be $c'_i \subseteq c'$. Assuming the assertion (3.b) of this Proposition does not hold, $c'_i \not\subseteq c$, for all $1 \leq i \leq r$; Note that each c'_i is a non-empty clause, since s is satisfiable. Furthermore, assume $c'_i = \neg d^i_1 \vee \ldots \vee \neg d^i_{t_i}$, for all $1 \leq i \leq v$, and let D denote the set $\{\neg d^i_j \,|\, 1 \leq i \leq v, 1 \leq j \leq t_i\}$. Note that D does not contain any negated atoms from A.

If (3.a) does not hold, then c is not comprised of any subset of clauses $\neg a_1, \ldots, \neg a_r$. Let's now construct a k-state $\gamma_0 = (w', \{w', w''\})$ which is a model of s, but not of a model of $KW(c)$.

1) $w'(a_i), w''(a_i)$, for all $1 \leq i \leq r$,
2) $w'(\neg d)$ and $w''(\neg d)$ for all atoms d such that $\neg d \in D$ and $\neg d \not\subseteq c$,
3) $w'(\neg e)$ and $w''(e)$ for any other atom e that has not been given a value.

γ_0 is a model of s, because it's a model for all $Kn(a_i)$, $1 \leq i \leq r$ due to step 1 in the construction above. γ_0 is a model of $Kn(c_j)$, for all $1 \leq j \leq n$, because for each c_j at least one of its disjuncts will be assigned a value by application of step 2. Furthermore, γ_0 is not a model of $KW(c)$, because all of atoms participating in c will be assigned a value by a combination of steps 1 or 3, and at least one atom of c will be assigned a value by step 3, resulting in $w'(c)$ and $w''(\neg c)$.

■

PROPOSITION 3.— $KW(c_1), \ldots, KW(c_n) \models_k KW(c)$ if and only if there exists a set of indices $i_1, \ldots, i_m$ where $1 \leq i_j \leq n$ for each i_j, such that $c = c_{i_1} \vee \ldots \vee c_{i_m}$.

PROOF. — The *if* part of the claims follows from the soundness of rule (R5).

We have to prove that $KW(c_1), \ldots, KW(c_n) \models_k KW(c)$ only if there is a subset of clauses $c_1, \ldots, c_n$ that together contain all and only disjuncts of c. Suppose there's no such subset, i.e. for all such subsets $\{c_{i_1}, \ldots, c_{i_m}\}$, $c_{i_1} \vee \ldots \vee c_{i_m} \neq c$. We will construct a model $\gamma_0 = (w', \{w', w''\})$ of $KW(c_1), \ldots, KW(c_n)$ that is not a model of $KW(c)$, thus contradicting the entailment. First, let's use C to denote the set $\{c_1, \ldots, c_n\}$, and consider those clauses from C that contain at least one disjunct that does not appear in c. Let's denote the set of these clauses C_1, and let A_1 denote the set of all atoms of $c_1, \ldots, c_n$ that do not appear in c. Let

$$w'(\neg d) \text{ and } w''(\neg d), \text{ for all } d \in A_1. \tag{11}$$

Thus, $\gamma_0 = (w', \{w', w''\})$ is a model of $KW(c_i)$ for all $c_i \in C_1$, since $w'(c_i)$ and $w''(c_i)$.

Next, consider the rest of the clauses from C, i.e. the set $C_2 = C \backslash C_1$. Let A_2 denote the set of all atoms of clauses in C_2. Let

$$w'(b) \text{ and } w''(b), \text{ for all } b \in A_2. \tag{12}$$

Thus, $\gamma_0 = (w', \{w', w''\})$ is a model of $KW(c_j)$ for all $c_j \in C_2$, since $w'(\neg c_j)$ and $w''(\neg c_j)$.

At this point, there are atoms of c whose values remain unassigned. Otherwise, if all atoms of c were assigned, since no atoms of A_1 appear in c it would mean that c consists entirely from atoms in A_2. Since each clause of C_2 is subsumed by c, it would follow that c equals to some subset of clauses in C_2. Let a be some still unassigned atom of c, and let

$$w'(a) \text{ and } w''(\neg a). \tag{13}$$

and let all the other unassigned atoms be assigned true in both w' and w'', i.e.

$$w'(e) \text{ and } w''(e), \text{ for all } e \neq a, e \notin A_2 \cup A_1. \tag{14}$$

At this point, worlds w', w'' and thus γ_0 are completely specified. γ_0 is not a model of $KW(c)$, because $w'(c)$, yet $w''(\neg c)$. ■

PROPOSITION 4.—Suppose $\{Kn(p_1), \ldots, Kn(p_m)\}$ is a satisfiable set. $Kn(p_1), \ldots, Kn(p_m) \models_k Kn(c)$ if and only if it can be derived using rules (R1) and (R2).

PROOF. — Note that rules (R1) and (R2) are analogous to unit clause resolution and subsumption rules of the domain language $\mathcal{L_D}$, i.e. every valid application of rules (r1), (r2) corresponds to a valid application of (R1) and (R2). According to Proposition 1 $Kn(p_1), \ldots, Kn(p_m) \models_k Kn(c)$ if and only if $p_1, \ldots, p_m \models c$. Furthermore, since the deduction from a satisfiable set in the domain language $\mathcal{L_D}$ is closed under unit clause resolution (r1) and subsumption (r2), $p_1, \ldots, p_m \models c$ if and only if there is a derivation of c from $p_1, \ldots, p_m$ using only these two rules. We can produce a valid derivation of $Kn(c)$ from $Kn(p_1), \ldots, Kn(p_m)$ from the derivation of c in the domain language by replacing $p_1, \ldots, p_m$ with $Kn(p_1), \ldots, Kn(p_m)$, replacing each application of (r1) with (R1), and replacing each application of (r2) with (R2).

■

PROPOSITION 5.— Suppose $\{Kn(p_1), \ldots, Kn(p_m)\}$ is a satisfiable set. $Kn(p_1), \ldots, Kn(p_m) \models_k KW(c)$ if and only if it can be derived using rules (R1), (R2), (R3), (R4) and (R5).

PROOF. — The *if*-part can be easily verified by verifying the soundness of each rule.

To prove the *only if* part we present below informal sketch of a derivation of $KW(c)$ from $s = \{Kn(p_1), \ldots, Kn(p_m)\}$ using the specified above rules of inference.

Observe that case 3 of Proposition 2 implies $Kn(p_1), \ldots, Kn(p_m) \models_k KW(c)$ if and only if

1) there exists a set of domain atoms $a_{i_1}, \ldots, a_{i_n}$, such that $Kn(a_{i_j}) \in s$ and $c = \neg a_{i_1} \vee \ldots \vee \neg a_{i_n}$, or

2) there exists a domain clause c', such that $s \models_k Kn(c')$ and $c' \subseteq c$.

If case number 1 above is true, there is a derivation of $KW(c)$ that consists of instances of (R3), deriving $KW(\neg a_j)$ from the premise $Kn(a_j)$ for each $a_j \in \{a_{i_1}, \ldots, a_{i_n}\}$, and instances of (R5) applied to the results of previous derivations, eventually deriving $KW(\neg a_{i_1} \vee \ldots \vee \neg a_{i_n})$, i.e $KW(c)$.

If case number 2 above is true, according to Proposition 4, $Kn(c')$ is derivable from s using rules (R1) and (R2). We can extend this derivation by applying (R2) to $Kn(c')$ to obtain $Kn(c)$, since $c' \subseteq c$. Furthermore, application of (R4) to $Kn(c)$ yields $KW(c)$.

■

PROPOSITION 6.— $\mathcal{B}(s) = \{w \,|\, \forall p. \in \mathcal{L}_{\mathcal{D}}(s \models_k Kn(p)) \Rightarrow w(p)\}$, or in other words, the set of possible worlds is equivalent to the set of all worlds in which everything known to the agent is true.

PROOF. — This proposition is easily verified by checking that (a) for every domain proposition p and SOK s, $s \models_k Kn(p)$ if and only if p is true in every possible world of $s, \mathcal{B}(s)$, and (b) every world w in which every proposition p such that $s \models_k Kn(p)$ is true, $w \in \mathcal{B}(s)$. ■

PROPOSITION 7.— $\mathcal{B}(s) = \mathcal{B}(CKWA(s))$. (Note that s is a set of ground *Kn*-atoms of $\mathcal{L}_{\mathcal{K}}$).

PROOF. — From the definition of $CKWA(s)$ it follows that $\mathcal{B}(CKWA(s)) \subseteq \mathcal{B}(s)$.

To prove that $\mathcal{B}(s) \subseteq \mathcal{B}(CKWA(s))$ consider an arbitrary k-state model $\gamma = (w, W)$ in $\alpha(s)$. If we show that any such $w \in \mathcal{B}(s)$ appears in some model γ' of $\mathcal{B}(CKWA(s))$, that would prove that $\mathcal{B}(s) \subseteq \mathcal{B}(CKWA(s))$. Given such γ, we construct γ' as follows. Let w_p denote a word state that is obtained from w by negating the value of the atom p, in other words, w_p differs from w only in the value of p. Let γ' be a k-state (w, W'), where $W' = \{w\} \cup \{w_p \,|\, p \text{ is an atom and } s \not\models_k KW(\neg p)\}$. γ' is a model of $CKWA(s) = s \cup \{\neg KW(p) \,|\, p \in \mathcal{L}_{\mathcal{D}} \wedge s \not\models_k KW(p)\}$, since w satisfies all domain propositions of s and, furthermore, for every ground proposition $\neg KW(\neg p) \in CKWA(s)$, there exist world states in W', namely w and w_p in which the truth values of p are different. Thus, γ' is a model of $CKWA(s)$, and therefore $w \in \mathcal{B}(CKWA(s))$.

■

LEMMA 8.— Let ψ and ψ' denote simple *Kn*-ψ-forms , and $\tilde{\psi}'$ denote a simple *KW*-ψ-form.

1) $\psi \models_k \psi'$ if and only if there exists a substitution σ such that $\mathcal{M}(\psi)\sigma \subseteq \mathcal{M}(\psi')$

2) $\psi \models_k \tilde{\psi}'$ if and only if $\psi \models_k \psi'$, where ψ' is obtained by replacing *KW* in $\tilde{\psi}'$ by *Kn*.

PROOF. — 1).*if* part. Let $Kn(c')$ be an arbitrary ground instance of ψ', i.e. $Kn(c') = \mathcal{M}(\psi)\sigma'$ for some σ'. Consider ground instance $Kn(c)$ of ψ, such that $Kn(c) = \mathcal{M}(\psi)\sigma\sigma'$. Since $\mathcal{M}(\psi)\sigma \subseteq \mathcal{M}(\psi')$ and $Kn(c') = \mathcal{M}(\psi')\sigma'$, therefore $c \subseteq c'$, and therefore by Proposition 1 $Kn(c) \models_k Kn(c')$. We have shown that for any ground instance $Kn(c')$ of ψ' there is a ground instance of ψ, which entails $Kn(c')$, thus proving $\psi \models_k \psi'$.

only if part. Suppose $\psi \models_k \psi'$, yet there is no substitution σ as described in the condition of the Lemma. We will show that this yields a contradiction by constructing a model of of ψ that is not a model of ψ'.

When ψ' is not a singleton, let $Kn(c')$ be an instance of $\mathcal{M}(\psi')$, which is obtained by assigning to each variable of ψ' a constant value which does not occur anywhere in $\mathcal{M}(\psi)$. Let's call these constant symbols *special* constants. Since the domain is infinite, special constants can always be found.

Suppose $c' = \neg d'_k \ldots \vee \ldots \neg d'_k$. Consider $\gamma_0 = (w, \{w'\})$ such that $w(d'_k)$ for every d'_k and $w(\neg a)$ for all other atoms. Clearly, $w(c')$ is false and therefore γ_0 is not a model of ψ'. On the other hand, unless at least one of ground instances of ψ consists entirely of a subset of literals $\neg d'_k$ of c', γ_0 would be a model of ψ. Suppose that ψ indeed defines a ground instance $c = \mathcal{M}(\psi)\sigma^0$ such that $c \subseteq c'$, but then it is possible to construct a substitution σ such that $\mathcal{M}(\psi)\sigma \subseteq \mathcal{M}(\psi')$, violating our assumption. σ is constructed from σ^0 by replacing each special constant by a variable of ψ', which is bound to that constant by σ^0. We have reached a contradiction.

The case of a singleton ψ', let $c' = \mathcal{M}(\psi')$ and repeat the argument of the above paragraph.

2) *if* part. For an arbitrary ground instance $KW(c')$ of $\tilde{\psi}'$ there exists a corresponding ground instance $Kn(c')$ of ψ', and $Kn(c') \models_k KW(c')$. Thus, $\psi \models_k \psi'$ implies $\psi \models_k \tilde{\psi}'$.

only if part. The proof is similar to the one from part 1. We derive a contradiction by constructing a model of of ψ that is not a model of $\tilde{\psi}'$.

When $\tilde{\psi}'$ is a singleton, the statement of the Lemma follows from part 1 of this Lemma and Proposition 1. When $\tilde{\psi}'$ is not a singleton, let $KW(c')$ be a ground instance of $\mathcal{M}(\tilde{\psi}')$ constructed by instantiating all variables of $\tilde{\psi}'$ with constant values which do not occur anywhere in $\mathcal{M}(\psi)$. Suppose $c' = \neg d'_k \ldots \vee \ldots \neg d'_k$. Consider $\gamma_0 = (w', \{w', w''\})$ such that

– $w'(d'_k)$ for every d'_k of c', and $w'(\neg a)$ for all other atoms, and
– $w''(\neg a)$ for all atoms a, including $d'_1, \ldots, d'_k$

Clearly, $w''(c')$ is false, while $w'(c')$ is true, and therefore γ_0 is not a model of $KW(c')$ and thus not a model of $\tilde{\psi}'$. From here on we reach a contradiction by applying the same argument as in part 1 of this proof.

■

THEOREM 9.— Given a set of *Kn*-ψ-forms $\Psi = \{\psi_1, \ldots, \psi_n\}$ and a *Kn*-ψ-form ψ, $\Psi \models \psi$ only if there is $\psi_i \in \Psi$ such that $([\mathcal{M}(\psi_i)] \models_k [\mathcal{M}(\psi)])$.

PROOF. — The proof is similar to the only-if proof from part 1 of Lemma 8. Assuming the statement of this theorem does not hold, we derive a contradiction by constructing a model of of ψ that is not a model of Ψ.

Suppose none of $[\mathcal{M}(\psi_1)], \ldots, [\mathcal{M}(\psi_n)]$ entail $[\mathcal{M}(\psi)]$. If ψ is a singleton, let's use $Kn(c)$ to denote the single ground instance of ψ, or, in case it's not a singleton, let $Kn(c)$ be a ground instance of $\mathcal{M}(\psi)$ constructed using a substitution σ instantiating all variables of ψ with constant values which do not occur anywhere in $\psi_1, \ldots, \psi_n$, nor in the exceptions of ψ. This is always possible due to infinite number of constants in the language. Suppose $c = \neg d_1 \vee \ldots \vee \neg d_k$ and consider $\gamma_0 = (w, \{w\})$ such that $w(d_k)$ for every d_k and $w(\neg a)$ for all other atoms. Since $w(c)$ is false, $\gamma_0 \notin \alpha(\psi)$. However, $\gamma_0 \in \alpha(\Psi)$, or otherwise there is a ground instance of some $\psi_i \in \Psi$, whose literals are a subset of $\{\neg d_1, \ldots, \neg d_k\}$ and we can construct a subset matching substitution from $\mathcal{M}(\psi_i)$ onto $\mathcal{M}(\psi)$(as done in the proof of part 1 of Lemma 8), which yields $([\mathcal{M}(\psi_i)] \models_k [\mathcal{M}(\psi)])$. We have arrived at a contradiction.

■

THEOREM 11.— Let $s = A \cup \Psi$ be a satisfiable saturated set of PSIPLAN-S *Kn*-atoms (A) and *Kn*-ψ-forms (Ψ), $s \models_k \psi$ if and only if

1) $\psi = [KW(\neg d_1 \vee \ldots \vee \neg d_m)]$ and $\{Kn(d_1), \ldots, Kn(d_m)\} \subseteq A$, or
2) there exists $\psi_k \in \Psi$, such that $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\psi)]$, and, furthermore,
$s - \{\psi_k\} \models_k (\psi \overset{e}{-} \psi_k)$

PROOF. — The *if*-part trivially follows from Proposition 2 and Definition 4.3 of e-difference.

The *only if* part. First, suppose ψ is a singleton $[KW(\neg d_1 \vee \ldots \vee \neg d_m)]$. Suppose that $A = \{Kn(a_1), \ldots, Kn(a_n)\}$ and $\{Kn(d_1), \ldots, Kn(d_m)\} \not\subseteq A$. Let $\gamma = (w', \{w', w''\})$ such that

– $w'(a_1), \ldots, w'(a_n)$, and $w'(\neg a)$ for any other atom a,
– $w''(a_1), \ldots, w''(a_n)$, $w''(d_1), \ldots, w''(d_m)$, and $w''(\neg a)$ for any other atom a.

Note that $w' \neq w''$. $\gamma \in \alpha(A \cup \Psi)$ unless there exists a ground $Kn(c) \in \Psi$ such that c consists of a subset of negated atoms $a_1, \ldots, a_n, d_1, \ldots, d_m$. Since $A \cup \Psi$ is satisfiable, in such case c must contain negated atoms from $d_1, \ldots, d_m$ that are different from $a_1, \ldots, a_n$. Since $A \cup \Psi$ is saturated, c cannot contain atoms from $a_1, \ldots, a_n$. Thus, $\gamma \in \alpha(A \cup \Psi)$ unless there exists a ground $Kn(c) \in \Psi$ such that c consists of a subset of negated atoms $d_1, \ldots, d_m$ that are different from $a_1, \ldots, a_n$.

If there is such a clause $Kn(c) \in \Psi$, then $Kn(c) \models_k Kn(\neg d_1 \vee \ldots \vee \neg d_m)$ and therefore $Kn(c) \models_k \psi$, thus $s \models_k \psi$ and it is easy to see that condition 2 of the theorem holds. Otherwise, $\gamma \in \alpha(A \cup \Psi)$ yet $\gamma \notin \alpha(\psi)$, because $\neg d_1 \vee \ldots \vee \neg d_m$ is true in w', but false in w''. We have reached a contradiction with the fact that $s \models_k \psi$.

Next, consider the case when ψ is not a singleton *KW*-ψ-form. If ψ is a singleton *Kn*-ψ-form, condition 2 of this Theorem trivially follows from Theorem 9. Finally, in case ψ is not a singleton, we first show the existence of ψ_k such that $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\psi)]$. Assuming there is no such $\psi_k \in s$, i.e. by Lemma 8 no $\psi_k \in s$ such that $\mathcal{M}(\psi_k)$ subset matches $\mathcal{M}(\psi)$, we construct a ground instance of ψ using a substitution that binds each variable of ψ to a unique constant which does not appear anywhere in the main form or exceptions of all ψ-forms of Ψ, ψ and atoms of A. Following a similar argument as the one used in the proof of Lemma 8, it is easy to show that this ground instance is not entailed by s unless there is a substitution σ that subset matches some $\mathcal{M}(\psi_k)$ onto $\mathcal{M}(\psi)$. Thus, there is $\psi_k \in \Psi$ such that $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\psi)]$. Furthermore, since the exceptions of ψ_k weaken it, the e-difference $\psi \overset{e}{-} \psi_k$ must be also entailed by s, i.e. it is necessary that $s - \{\psi_k\} \models_k (\psi \overset{e}{-} \psi_k)$. ■

LEMMA 13.— Let $\tilde{\psi}$, $\tilde{\psi}'$ denote simple *KW*-ψ-forms without repeated predicate symbols in the main form. $\tilde{\psi} \models_k \tilde{\psi}'$ if and only if there exists a substitution σ such that $\mathcal{M}(\tilde{\psi})\sigma = \mathcal{M}(\tilde{\psi}')$.

PROOF. — *if* part proof trivially follows from Proposition 3.

only if part. When $\tilde{\psi}'$ is a singleton, given that $\tilde{\psi}$, $\tilde{\psi}'$ have no repeated predicate symbols, Proposition 3 implies there is a single ground instance of $\tilde{\psi}$ that equals the singleton $\tilde{\psi}'$ thus proving the statement of this Lemma.

When $\tilde{\psi}'$ is not a singleton, the proof is by contradiction. We pick a ground instance $KW(c)$ of $\tilde{\psi}'$ by instantiating all variables of $\tilde{\psi}'$ to different constant values that do not appear in $\tilde{\psi}'$ and $\tilde{\psi}$. Assuming the statement of the Lemma is not true, we show that such $KW(c)$ is not entailed by $\tilde{\psi}$ by constructing a model γ of $\tilde{\psi}$ that is not a model of $KW(c)$, unless there is a ground instance of $\tilde{\psi}$ that is the same as $KW(c)$, in which case a substitution σ such that $\mathcal{M}(\tilde{\psi})\sigma = \mathcal{M}(\tilde{\psi}')$ can be found.

Let $\gamma = (w', \{w', w''\})$, where

– $w'(d), w''(d)$, for those atoms d that appears both in c and in some ground instance of $\tilde{\psi}$.

At this point not all of the atoms of c are assigned values by w', w''. Otherwise, it would follow that there exist ground instances of $\tilde{\psi}$ that together contain all atoms of c. Furthermore, since $\tilde{\psi}, \tilde{\psi}'$ do not have any repeated predicate symbols, there exists a single ground instance of $\tilde{\psi}$ equal to $KW(c')$ and therefore we could construct a substitution σ such that $\mathcal{M}(\tilde{\psi})\sigma = \mathcal{M}(\tilde{\psi}')$ as done in the end of proof of part 1, Lemma 8.

Thus, there exists at least one atom of c that does not appear in any ground instance of $\mathcal{M}(\tilde{\psi})$. Let's call it a and set

– $w'(a), w''(\neg a)$, for some atom a that appears in c and does not appear in any ground instance of $\tilde{\psi}$.

Furthermore, let's set

– $w'(\neg d), w''(\neg d)$, for every atom d that appears in some ground instance of $\tilde{\psi}$, but does not appear in c, and

– $w'(d), w''(d)$, for all remaining atoms.

It is easy to see that $w'(c)$ and $w''(\neg c)$ due to the first two steps of our construction, i.e. $\gamma \notin KW(c)$, thus $\gamma \notin \alpha(\tilde{\psi}')$. On the other hand, $\gamma \in \alpha(\tilde{\psi})$, because w' differs from w'' only in the value of a single atom a, which is not found in any ground instances of $\tilde{\psi}$. ■

THEOREM 16.— *Let A denote a set of Kn-atoms $\{Kn(a_1), \ldots, Kn(a_k)\}$, Ψ denote a set of Kn-ψ-forms $\{\psi_1, \ldots, \psi_r\}$ and $\tilde{\Psi}$ denote a set of KW-ψ-forms $\{\tilde{\psi}_1, \ldots, \tilde{\psi}_t\}$. Each KW-$\psi$-form of $\tilde{\Psi}$ does not have repeated predicate symbols in its main form.*

Assume $A \cup \Psi \cup \tilde{\Psi}$ is satisfiable and saturated. Let $\tilde{\psi}$ be a KW-ψ-form without repeated predicate symbols in the main form. Then, $A \cup \Psi \cup \tilde{\Psi} \models_k \tilde{\psi}$ if at least one of the following conditions holds:

1) There are $Kn(a_1), \ldots, Kn(a_n) \in A$, such that $\tilde{\psi} = [KW(\neg a_1 \vee \ldots \vee \neg a_n)]$.

2) There is a $\psi_k \in \Psi \cup \tilde{\Psi}$ such that $[\mathcal{M}(\psi_k)] \models_k [\mathcal{M}(\tilde{\psi})]$, and $A \cup \Psi \cup \tilde{\Psi} - \psi_k \models_k \tilde{\psi} \overset{e}{-} \psi_k$.

3) There is a decomposition of the main part of $\tilde{\psi}$ into two subclauses Q' and Q'' such that $\mathcal{M}(\tilde{\psi}) = Q' \vee Q''$ and

$$A \cup \tilde{\Psi} \models_k [KW(Q') \,|\, \Sigma'], \quad \textit{and}$$
$$A \cup \tilde{\Psi} \models_k [KW(Q'') \,|\, \Sigma''],$$

where Σ', Σ'' are subsets of the set of exceptions of $\tilde{\psi}$, which only use variables from Q' and Q'', respectively.

4) Assume $\tilde{\psi} = [KW(Q)$ except $\Sigma]$. There is an atom $Kn(a_i) \in A$ and $\tilde{\psi}' \in \tilde{\Psi}$ such that $\tilde{\psi}' \models_k [KW(\neg a_i \vee Q)$ except $\Sigma]$.

PROOF. — Cases 1 and 2 follow directly from Theorem 11 and Lemma 13.

Case 3. If there is such a decomposition $\mathcal{M}(\tilde{\psi}) = Q' \vee Q''$, then for any ground *KW*-proposition $c \in \tilde{\psi}$, obtained by instantiating the main form $\mathcal{M}(\tilde{\psi})$ with ground substitution σ, we have that $c = Q'\sigma \vee Q''\sigma$, and since $A \cup \tilde{\Psi}$ entails $KW(Q'\sigma)$ and $KW(Q''\sigma)$, by rule (R6) we conclude that $A \cup \Psi \models_k c$, and thus $A \cup \Psi \models_k \tilde{\psi}$.

Case 4. The proof is similar to the one for case 3. For every ground instance c of $\tilde{\psi}$, there is a ground instance of $\tilde{\psi}'$, which together with $Kn(a_i)$ implies c by application of rule (R6). ■

THEOREM 18.— Assume s is satisfiable and saturated. The following state of knowledge update procedure is correct and complete:

$$update(s, a) = \begin{cases} ((s \overset{e}{-} \mathcal{A}^-(a)) \overset{e}{-} \mathcal{A}(a)) \cup \mathcal{A}(a), & \text{when } a \text{ is a domain action,} \\ s \cup \Delta(a), & \text{when } a \text{ is a sensing action.} \end{cases}$$

PROOF. — Proving correctness and completeness requires showing $do(a, \mathcal{B}(s \cup \Delta(a))) = \mathcal{B}(update(s \cup \Delta(a), a))$. Since for a domain action $\Delta(a) = \emptyset$, the case of update for the domain action is reduced to showing $do(a, \mathcal{B}(s \cup \Delta(a))) = \mathcal{B}(update(s \cup \Delta(a), a))$. That proof is presented in (Babaian *et al.*, 2006).

The case of update after a sensing action is straightforward, given that the possible world transition model for a sensing action a_s is defined as $do(a_s, W) = W$. ■