Comparing Answer Set Programming and Hierarchical Knowledge Bases Regarding Comprehensibility and Reasoning Efficiency in the Context of Agents

Corinna Krüger, Daan Apeldoorn, Gabriele Kern-Isberner
Technische Universität Dortmund
{corinna.krueger, daan.apeldoorn}@tu-dortmund.de, gabriele.kern-isberner@cs.tu-dortmund.de

Abstract

In this paper, answer set programming and hierarchical knowledge bases are compared as knowledge representation paradigms for representing agent behavior. The comparison is based on two evaluation criteria: (1) comprehensibility (i. e., how easily the represented agent behavior can be comprehended by humans) and (2) reasoning efficiency (i. e., which of the two paradigms allows agents for more efficient reasoning about their next actions). It is shown that hierarchical knowledge bases seem to be the more comprehensible and more efficient approach for implementing agent behavior.

1 Introduction

Answer Set Programming (ASP) is a well known knowledge representation paradigm (based on *normal logic programs* with *default negation*, cf. (Gelfond and Lifschitz 1991)), which is nowadays widely used both in theory and practice. With efficient solvers like CLINGO[1] and DLV[2], especially the practical aspect of ASP became obvious. Nevertheless, when it comes to a larger knowledge base, the representation as an answer set program can become unhandy and the extensive use of default negation can make the represented knowledge hard to comprehend. One reason for that is, that the knowledge contained in an answer set program is basically a set of rules which does not allow for the explicit expression of cognitive concepts like *abstraction* and *generalization*. Such concepts, however, are extremely important for humans to be able to work cognitively with a larger amount of knowledge.

In this paper, we compare ASP to an alternative approach called *Hierarchical Knowledge Bases* (HKBs) according to (Apeldoorn and Kern-Isberner 2017)[3] which were originally designed to represent knowledge gained by learning agents. After a discussion of related work (Section 2), ASP and HKBs will be briefly outlined (Section 3). After that, a survey will be described which was performed to compare both approaches regarding their comprehensibility (Section 4.1), and the reasoning efficiency of both approaches will be evaluated in the context of agents (Section 4.2). We show that in our experiments, HKBs outperform ASP both regarding comprehensibility and reasoning efficiency. Finally, a conclusion on our results will be provided (Section 5).

2 Related Work

This paper compares two knowledge representation formalisms, namely the well established ASP and the rather novel approach of HKBs according to (Apeldoorn and Kern-Isberner 2017): The former has its origins in the late 1980s / early 1990s (cf. (Gelfond and Lifschitz 1991)) and is nowadays a well-established paradigm for the rule-based representation of knowledge. Furthermore, several efficient solvers exist (e. g., CLINGO and DLV) which are able to calculate answer sets (i. e., inferred knowledge) efficiently from a given answer set program.

In (Apeldoorn and Kern-Isberner 2016) and (Apeldoorn and Kern-Isberner 2017) HKBs were designed to represent knowledge learned by an agent through sub-symbolic machine learning approaches: The main idea of such a HKB is, to represent rule-based knowledge on different levels of abstraction, were the topmost level contains the most general rule(s) and the lower levels contain more specific rules. In contrast, the HKB approach by (Borgida and Etherington 1989) is more geared towards relational-based data management and class hierarchies.

First ideas of incorporating a ranking into default-based approaches have been introduced in the late 1980s by Gerhard Brewka and were afterwards continued by others (see (Lang 2015) for a summary). According to these ideas, priorities are assigned to default rules to induce a ranking among these rules, which can improve the quality of inference results. However, HKBs have another focus and were mainly developed with the comprehensibility of a larger amount of knowledge in mind: Originally designed for the purpose of reflecting the knowledge learned by agents through machine learning approaches, HKBs represent knowledge compactly in a top-down manner. The contained knowledge can be read on an appropriate level of detail, following the idea of an hierarchical organization of knowledge (which is known to be an adequate cognitive model for the organization of knowledge, according to common psychological literature, e. g.,

[1] `https://sourceforge.net/projects/potassco/files/clingo/` (last accessed on 2016-12-17)

[2] `http://www.dlvsystem.com/dlv/` (last accessed on 2016-04-26)

[3] Not to be confused with the *Hierarchical Knowledge Base* approach by Borgida et al. (Borgida and Etherington 1989).

(Zimbardo and Gerrig 2004), pp. 328–331).

Nevertheless, the human comprehensibility aspect of knowledge representation in the context of agents seems to be rarely studied so far.

3 Preliminaries

This section describes the preliminaries needed for the comparison which will be done later in Section 4. In the first place, the considered agent model will be introduced (Section 3.1). After that, the concepts of answer set programs and HKBs will be briefly described in the context of the introduced agent model (Sections 3.2 and 3.3).

3.1 Agent Model and Environment

As a general agent model for our experiments, we consider an agent which is equipped with n Sensors. Every sensor represents one dimension $\mathbb{S}_i$ of the agent's state space, where every $\mathbb{S}_i$ is a set of discrete sensor values. Furthermore, the agent is able to perform actions which are selected from a predefined action set $\mathbb{A}$.

More concretely, for our experiments, we consider discrete two-dimensional grid world scenarios, where an agent has to navigate from a starting point to a destination. The agent is equipped with two sensors, an x and a y sensor, to determine its position in the environment with the corresponding sensor value sets $\mathbb{S}_x = \{s_1^x, ..., s_k^x\}$ and a $\mathbb{S}_y = \{s_1^y, ..., s_l^y\}$. Furthermore, the agent is able perform actions from the action set $\mathbb{A} = \{\text{North}, \text{South}, \text{East}, \text{West}\}$. By performing an action $a \in \mathbb{A}$, the agent moves one cell into the respective direction of the grid world.

3.2 Answer Set Programs

In the following, the basic ideas of an answer set program will be briefly summarized, mainly following (Kern-Isberner and Beierle 2014), with a focus on the aspects which are needed for our experiments:

An answer set program, in its basic form, can be considered a set of rules $\mathcal{P} = \{r_1, ..., r_m\}$, where every rule $r \in \mathcal{P}$ is of the form $H \leftarrow A_1, ..., A_o, not\ B_1, ..., not\ B_p$. H is the rule's *head* and $A_1, ..., A_o, not\ B_1, ..., not\ B_p$ is the rule's *tail*. All elements $H, A_1, ..., A_o, B_1, ..., B_p$ are literals and not is the default negation operator (with the meaning that $not\ X$ evaluates to $true$, if X is either *false* or X is not known in $\mathcal{P}$). The head of a rule can be inferred from the rule's tail, if the tail evaluates to $true$. Moreover, rules can have an empty tail. Such rules are called *facts* in the program and can be inferred without any other literals being involved.

To infer knowledge from an answer set program, the basic idea is to collect the heads of all rules whose tails can be evaluated to $true$, given the program's facts together with the heads of all other rules which can be inferred. Algorithms to compute such so-called *answer sets* are well-known and implemented in established solvers like CLINGO and DLV.

Example 1 *We consider an agent according to the model described in Section 3.1 which is equipped with an x and a y sensor to determine its position in a two-dimensional grid world environment (see Figure 1). The agent has to navigate from a starting point A to a target point B. A possible answer set program to describe the knowledge needed by the agent to get from the starting point A to the target point B could look as shown in Figure 2 (a).*

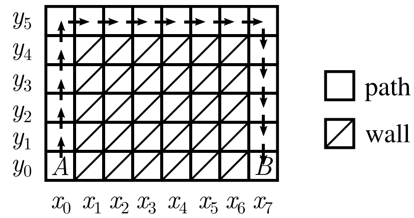


Figure 1: Agent Navigation Task in a Grid World

(a)

$$\begin{aligned}\mathcal{P}_1 = \{\ & \text{East} \leftarrow not\ x_0, not\ x_7.,\\ & \text{North} \leftarrow x_0, not\ y_5.,\\ & \text{South} \leftarrow x_7.,\\ & \text{East} \leftarrow x_0, y_5.\}\end{aligned}$$

(b)

$\mathcal{KB}_1$:

$\top \Rightarrow$ East	R_1
$x_0 \Rightarrow$ North $x_7 \Rightarrow$ South	R_2
$x_0 \wedge y_5 \Rightarrow$ East	R_3

Figure 2: Knowledge Bases for the Grid World Navigation Task from Figure 1 as Answer Set Program (a) and HKB (b)

3.3 Hierarchical Knowledge Bases

In contrast to answer set programs, the knowledge contained in a HKB is represented in a top-down manner, starting with the most general knowledge on the top most level. In the following, a purly quantitative version of HKBs will be defined, mainly following (Apeldoorn and Kern-Isberner 2017).[4]

In the first place, *complete* and *partial states* as well as *complete* and *generalized rules* are introduced, which will be needed for the definition of a HKB.

Definition 1 (Complete States/Partial States) *A complete state is a conjunction $s := s_1 \wedge ... \wedge s_n$ of all values s_i currently perceived by an agent's sensors, where n is the number of sensors (and every perceived sensor value $s_i \in \mathbb{S}_i$ of the corresponding sensor value set $\mathbb{S}_i$ is assumed to be a fact in the agent's current state). A partial state is a conjunction $s := \bigwedge_{s' \in S} s'$ of a subset $S \subset \{s_1, ..., s_n\}$ of the sensor values of a complete state.*

Definition 2 (Complete Rules/Generalized Rules) *Both complete rules and generalized rules are of the form $p_\rho \Rightarrow$*

[4]Note that in the original definitions from (Apeldoorn and Kern-Isberner 2017), the rules contained in a HKB are weighted by real-valued weights and are therefore not (purely) quantitative. However, to preserve the comparability with classical answer set programs, HKBs will be introduced with pure quantitative rules here.

a_ρ, where p_ρ is either a complete state (in case of a complete rule) or a partial state (in case of a generalized rule) and the conclusion $a_\rho \in \mathbb{A}$ is an action of an agent's action space $\mathbb{A}$.

Thus, complete rules map complete states to actions and generalized rules map partial states to actions.

Based on the definitions of complete rules and generalized rules, an HKB is now defined as follows:

Definition 3 (Hierarchical Knowledge Base) *A Hierarchical Knowledge Base (HKB) is an ordered set $\mathcal{KB} := \{R_1, ..., R_{n+1}\}$ of $n + 1$ rule sets, where n is the number of sensors (i. e., the number of state space dimensions). Every set $R_{i<n+1}$ contains generalized rules and the set R_{n+1} contains complete rules, such that every premise $p_\rho = \bigwedge_{s \in S_\rho} s$ of a rule $\rho \in R_i$ is of length $|S_\rho| = i - 1$.*

According to Definition 3, the set R_1 contains the most general rules (with empty premises) and the set R_{n+1} contains the most specific (i. e., the complete) rules.

On every level $R_{j>1}$, the rules can be considered the exceptions of the more general rules on level R_{j-1}. By this, default negation (as known from ASP) is expressed implicitly.

Example 2 *We consider again the agent from Example 1 in Section 3.2 (see Figure 1). A possible HKB to describe the knowledge needed by the agent to get from the starting point A to the target point B could look as shown in Figure 2 (b).*

4 Comparison

In this section both knowledge representation paradigms (ASP and HKBs) will be compared regarding their comprehensibility by humans and their reasoning efficiency in the context of an agent.

To compare the comprehensibility, a survey based on a questionnaire has been performed which will be described in the following (Section 4.1). To compare the reasoning efficiency, the *average total time* of all reasoning operations needed by an agent to solve different grid world scenarios has been measured and compared under increasing scenario size and complexity (Section 4.2).

4.1 Comprehensibility

To measure and compare the comprehensibility of both knowledge representation paradigms, we performed a questionnaire-based survey in the context of a computer science lecture related to knowledge representation at the *TU Dortmund University*. The lecture was mainly dedicated to students in the fifth semester and it was ensured that at the time of the survey, ASP (as part of the lecture) was not yet introduced. Thus, all participants could be considered of having the educational requirements of understanding and working with knowledge representation approaches, but were not yet biased too much towards certain representation paradigms.

Questionnaire The survey was realized with a questionnaire which comprised the following four parts:

1. All participants received a written introduction with general explanations about the contents of the questionnaire and both knowledge representation paradigms (ASP and HKBs) where briefly introduced.
2. An introductory example including a small knowledge base of every paradigm with corresponding explanations was provided to the participants.
3. Two different scenario of identical size and complexity were introduced to the participants, where a robot has to navigate in a grid world environment (similar to the one described in Figure 1). The robot's knowledge bases were provided in ASP format for the first scenario and in HKB format for the second scenario.
4. In this part (which was the main part of the questionnaire), participants had to answer four questions per knowledge representation paradigm about what the robot would infer in general and, when being provided with certain sensor values. In addition, the participants were asked whether they favor one of the two knowledge representation paradigms and were able to underpin their opinion by providing advantages and disadvantages through answering to corresponding open question. Finally, the participants were asked whether they already knew one of the two approaches (these participants were later excluded in the evaluation).

To avoid a preference effect regarding the order in which the two knowledge representation paradigms were introduced, two different questionnaires were created with a permuted introduction order (one in which ASP was introduced first and another in which HKBs were introduced first).

Evaluation The initial sample size of our survey was $N = 57$ students. However, to avoid biasing, we did not consider persons, who were—at the time of the survey—already aware of one of the two knowledge representation paradigms (but not of both). After this preprocessing phase, the final sample size comprised $N' = 48$ students. [5]

To measure the comprehensibility from an *objective* point of view, we considered the mistakes (i. e., wrong inferences) made by the participants throughout the four questions per knowledge representation paradigm, since mistakes usually serve as an indicator for misunderstanding or a lack of understanding. The results are provided in Figure 3.

As can be seen in Figure 3, participants provided more wrong inferences using ASP compared to the HKB approach. However, in general, only a small amount of mistakes have been made by the participants and the difference of made mistakes between both approaches is rather small (which could possibly be caused by too simple scenarios or questions).[6]

[5]Although $N' = 48$ is a rather small sample size, the following results show a tendency of how people rate the comprehensibility of ASP vs. HKB. A further survey including more participants could be reasonable future work here.

[6]Note that this also contradicts some results of a preliminary

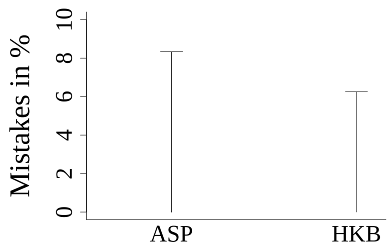


Figure 3: Bar Plot of Wrong Inferences Made by Participants with the ASP and the HKB Approach

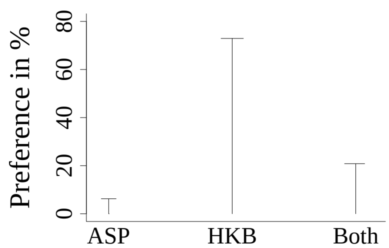


Figure 4: Barplot of the Preferences of the Participants for the Knowledge Representation Paradigms

To measure the comprehensibility from a *subjective* point of view, participants were asked which of the two approaches appears to be more comprehensible to them. The results are shown in Figure 4.

Figure 4 shows that about 70% prefer the HKB approach. Less than one tenth of the test persons were favoring ASP regarding comprehensibility. About 20% could not decide for one of the two approaches. Thus, it is obvious that HKBs seem to be much more comprehensible to the participants.

As mentioned before in the context of the description of the questionnaire, the participants were also able to underpin their opinion on the comprehensibility by providing advantages and disadvantages as answers to open questions. In the following, a brief summary of the most relevant answers will be provided:

ASP

(+) ASP seems to be closer to natural language because of the explicit default negation operator *not*. Another mentioned advantage is the possibility to express nescience. Furthermore, it seems to be easy to delete or to add rules. In addition, some participants clearly outlined that ASP rules are easy to handle.

(–) In contrast to that, ASP also appears to be confusing, complicated and the rules are built up against the usual reading direction. Moreover, the time for evaluation was criticized as the user always has to consider all rules to find the next action of the agent. Some participants were also remarking that meaning of *not* is hard to understand.

HKBs

(+) Concerning the HKB approach, the clarity, compactness and structure of HKBs is regarded as positive and HKBs appear to be intuitive. As a further advantage, it was mentioned that the rules are shorter and are conform to the common reading direction. The presence of a general "default rule" appeared to be useful as well. The time need for evaluation was also discussed in the context of the HKB approach: Since it is not always necessary to consider all rules, this is considered to have a positive effect on the reasoning time. Furthermore, it appears to have a positive effect for the understanding of the knowledge that a hierarchical structure exists and no explicit negation is needed.

(–) As disadvantages, it is remarked, that both the negation and the nescience can't be represented explicitly using the HKB approach. In addition, HKBs are criticized for being searched for the correct rule from bottom to top (which seems to appear contra-intuitive). Furthermore, the assumption was made, that it could be difficult to add knowledge to an HKB.

Even if it seems to be a matter of personal taste according to the diverse opinions, which approach is preferred, it is also clear that in our survey the HKB approach is clearly preferred over ASP under the aspect of comprehensibility.

4.2 Reasoning Efficiency

In this section both knowledge representation paradigms (ASP and HKBs) will be compared regarding their reasoning efficiency in the context of an agent. For this purpose, the total time of all reasoning operations needed by an agent to solve different grid world scenarios was measured and compared. In every grid world scenario the agent has to navigate from a starting point A to a target point B (similar to the example provided in Figure 1).

To measure the needed time to solve a grid world based on an answer set program, the state-of-the-art ASP solver CLINGO is used. To solve a grid world scenario based on HKBs, the HKB reasoning algorithm according to (Apeldoorn and Kern-Isberner 2016) is used.[7]

To discover possible differences regarding the efficiency of both representation paradigms we run two different test series:

survey with an earlier version of the questionnaire and a smaller but more heterogeneous sample of students from different fields of study (cf. (Krüger 2016)): In this survey, participants made slightly more mistakes using HKBs; the results for the favored approach were comparable, however.

[7]Given a premise, this algorithm searches a HKB upwards starting from the bottom-most level R_{n+1} for the first rule whose premise is fulfilled and returns its concluding action (see (Apeldoorn and Kern-Isberner 2016) for details).

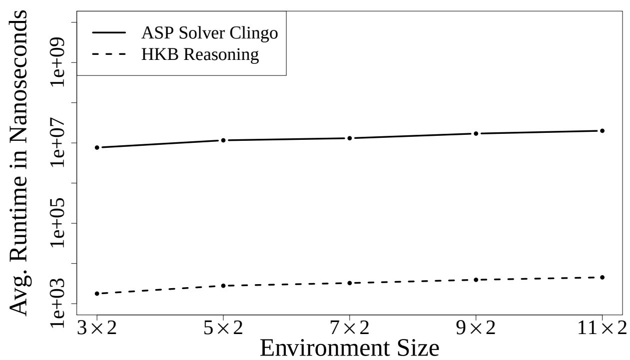


Figure 5: Runtime Comparison of HKB Reasoning and ASP Solver CLINGO for Increasing Environment Sizes

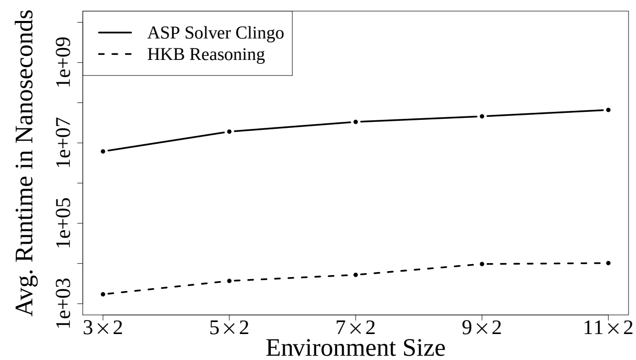


Figure 6: Runtime Comparison of HKB Reasoning and ASP Solver CLINGO for Increasing Environment Size and Complexity

- The first test series comprises five scenarios of growing size: Theses scenarios only differ in the size of the environment and are identical regarding their structural complexity. This leads to more calls to the respective reasoner but the knowledge base representing the agent's knowledge is the same for all scenarios, see Figure 7 (a).
- The second test series comprises five scenarios of growing size *and* structural complexity, which results not only in more calls to the reasoner but also in larger knowledge bases (reflecting the structural increasing complexity of the scenarios), see Figure 7 (b).

In the following both test series will be explained in detail and the results will be discussed.

Increasing Environment Size Only In the first test series, we analyze how an increasing environment size will affect the reasoning efficiency of both the ASP and the HKB approach. For this purpose, five navigation scenarios are considered, in which the agent has to navigate from a starting point A to a target point B, avoiding a wall in the south of the grid world. We start with an environment of size 3×2 and the x dimension of the scenario will then successively be increased by 2 up to a size of 11×2. Figure 5 shows the results of the first test series: The x-axis represents the increasing size of the grid world, the y-axis represents the sum of reasoning time needed for all reasoning calls to solve the respective scenario in nanoseconds.[8] As one can easily see, according to Figure 5, the HKB reasoning algorithm clearly outperforms ASP reasoning through the state-of-the-art ASP solver CLINGO. One reason for that seems to be the fact that the ASP solver has to consider all rules, whereas the HKB reasoning algorithm only has to consider all rules in the worst case (i. e., if the only rule whose premise matches a perceived state is the rule on level R_1 and thus the algorithm has to search through all rules on all levels).

Note that the answer set programs used in our test series were automatically derived from the corresponding HKBs, which usually does not lead to the most compact ASP representations of the knowledge needed to solve the task. However, we do not assume the large gap of $\approx 10^4$ nanoseconds between the two reasoning algorithm that can be seen in Figure 5 to be closed by a slightly more compact ASP representation with a few less unnecessary rules.

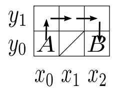
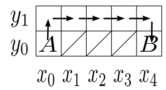
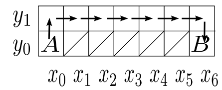
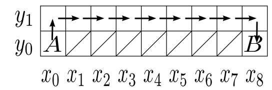
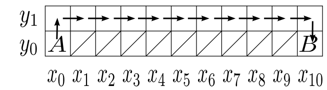
[8]Note that a logarithmic y-axis is used to be able to reflect the large difference between the performance of the ASP and the HKB approach.

Increasing Environment Size and Complexity In the second test series, we analyze how an increasing environment size with increasing structural complexity of the scenarios will affect the reasoning efficiency of both the ASP and the HKB approach. For this purpose, again, five navigation scenarios are considered, in which the agent has to navigate from a starting point A to a target point B. This time, the agent has to go around an increasing number of small walls located both in the north and the south of the grid world. We start again with an environment of size 3×2 and the x dimension of the scenario will then be successively increased by steps of 2 up to a size of 11×2.

Figure 6 shows the results of the second test series (the axes are assigned analogously to Figure 5): Also in this test series, the HKB reasoner clearly outperforms the ASP solver. Compared to Figure 5, in Figure 6 both curves have slightly higher slopes. Moreover, the ASP solver seems to behave slightly worse regarding the growing structural complexity of the scenarios. This effect can be explained by the steadily growing number of rules that need to be considered by the ASP reasoning algorithm in case the structural complexity of the scenarios increases (the HKB reasoning algorithm, in contrast, only has to consider all rules in the worst case).

Both test series indicate that agents navigating through grid world scenarios are significantly more efficient when representing their knowledge using HKBs instead of using ASP. An essential reason for the large difference in the runtime could be that, in case of ASP, usually all rules of the logical program need to be considered. In contrast to that, the reasoning algorithm for HKBs will stop searching, once an applicable rule was found: Only in the worst case

(a)

Env.					
HKB	$\mathcal{KB}_1$: $\top \Rightarrow$ East R_1 $y_0 \Rightarrow$ North $x_2 \Rightarrow$ South R_2	$\mathcal{KB}_2$: $\top \Rightarrow$ East R_1 $y_0 \Rightarrow$ North $x_4 \Rightarrow$ South R_2	$\mathcal{KB}_3$: $\top \Rightarrow$ East R_1 $y_0 \Rightarrow$ North $x_6 \Rightarrow$ South R_2	$\mathcal{KB}_4$: $\top \Rightarrow$ East R_1 $x_8 \Rightarrow$ North $y_0 \Rightarrow$ South R_2	$\mathcal{KB}_5$: $\top \Rightarrow$ East R_1 $y_0 \Rightarrow$ North $x_{10} \Rightarrow$ South R_2
ASP	$\mathcal{P}_1 := \{$East $\leftarrow$ *not* y_0, *not* x_2., North $\leftarrow y_0$., South $\leftarrow x_2$.$\}$	$\mathcal{P}_2 := \{$East $\leftarrow$ *not* y_0, *not* x_4., North $\leftarrow y_0$., South $\leftarrow x_4$.$\}$	$\mathcal{P}_3 := \{$East $\leftarrow$ *not* y_0, *not* x_6., North $\leftarrow y_0$., South $\leftarrow x_6$.$\}$	$\mathcal{P}_4 := \{$East $\leftarrow$ *not* y_0, *not* x_8., North $\leftarrow y_0$., South $\leftarrow x_8$.$\}$	$\mathcal{P}_5 := \{$East $\leftarrow$ *not* y_0, *not* x_{10}., North $\leftarrow y_0$., South $\leftarrow x_{10}$.$\}$

(b)

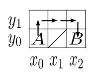
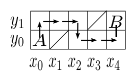
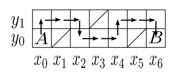
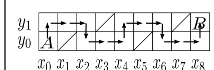
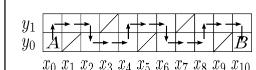
Env.					
HKB	$\mathcal{KB}_1$: $\top \Rightarrow$ East R_1 $y_0 \Rightarrow$ North $x_2 \Rightarrow$ South R_2	$\mathcal{KB}_2$: $\top \Rightarrow$ East R_1 $x_4 \Rightarrow$ North R_2 $x_2 \wedge y_1 \Rightarrow$ South $x_0 \wedge y_0 \Rightarrow$ North R_3	$\mathcal{KB}_3$: $\top \Rightarrow$ East R_1 $x_6 \Rightarrow$ South R_2 $x_0 \wedge y_0 \Rightarrow$ North $x_4 \wedge y_0 \Rightarrow$ North $x_2 \wedge y_1 \Rightarrow$ South R_3	$\mathcal{KB}_4$: $\top \Rightarrow$ East R_1 $x_8 \Rightarrow$ North R_2 $x_6 \wedge y_1 \Rightarrow$ South $x_4 \wedge y_0 \Rightarrow$ North $x_2 \wedge y_1 \Rightarrow$ South $x_0 \wedge y_0 \Rightarrow$ North R_3	$\mathcal{KB}_5$: $\top \Rightarrow$ East R_1 $x_{10} \Rightarrow$ South R_2 $x_8 \wedge y_0 \Rightarrow$ North $x_6 \wedge y_1 \Rightarrow$ South $x_4 \wedge y_0 \Rightarrow$ North $x_2 \wedge y_0 \Rightarrow$ South $x_0 \wedge y_0 \Rightarrow$ North R_3
ASP	$\mathcal{P}_1 : \{$East $\leftarrow$ *not* y_0, *not* x_2., North $\leftarrow y_0$., South $\leftarrow x_2$.$\}$	$\mathcal{P}_2 : \{$East $\leftarrow$ *not* $x_4, \{x_0, y_0\}1, \{x_2, y_1\}1$., North $\leftarrow x_4, \{x_0, y_0\}1, \{x_2, y_1\}1$., North $\leftarrow x_0, y_0$., South $\leftarrow x_2, y_1$.$\}$	$\mathcal{P}_3 : \{$East $\leftarrow$ *not* $x_6, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_0, y_0\}1$., South $\leftarrow x_6, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_0, y_0\}1$., North $\leftarrow x_0, y_0$., South $\leftarrow x_2, y_1$., North $\leftarrow x_4, y_0$.$\}$	$\mathcal{P}_4 : \{$East $\leftarrow$ *not* $x_8, \{x_0, y_0\}1, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_6, y_1\}1$., North $\leftarrow x_8, \{x_0, y_0\}1, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_6, y_1\}1$., North $\leftarrow x_0, y_0$., South $\leftarrow x_2, y_1$., North $\leftarrow x_4, y_0$., South $\leftarrow x_6, y_1$.$\}$	$\mathcal{P}_5 : \{$East $\leftarrow$ *not* $y_{10}, \{x_8, y_0\}1, \{x_0, y_0\}1, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_6, y_1\}1$., South $\leftarrow x_{10}, \{x_8, y_0\}1, \{x_0, y_0\}1, \{x_2, y_1\}1, \{x_4, y_0\}1, \{x_6, y_1\}1$., North $\leftarrow x_8, y_0$., North $\leftarrow x_0, y_0$., South $\leftarrow x_2, y_0$., North $\leftarrow x_4, y_0$., South $\leftarrow x_6, y_1$.$\}$

Figure 7: Grid World Scenarios of Growing Size (a) and Size and Complexity (b): In case of (b), answer set programs make use of the CLINGO expression $\{x, y\}1$, which evaluates to *true* if at most one of the literals x and y is *true*. This merely serves as a shortcut to represent the answer set programs more compactly.

all rules of the knowledge base have to be to considered. Although the answer set programs were in general not minimal, we do not expect this to affect our results substantially.

5 Conclusion and Future Work

In this paper, ASP and HKBs were compared as knowledge representation paradigms for agents in terms of comprehensibility and reasoning efficiency.

First, we gave a brief introduction of the considered agent model and its environment, as well as for ASP and HKBs. Subsequently, the experiments and the comparison of the two approaches regarding both comprehensibility and reasoning efficiency have been described. To compare the comprehensibility of both knowledge representation paradigms, a questionnaire-based survey had been designed which was realized during a knowledge representation related lecture at the *TU Dortmund University*. Thus, most of the participants were undergraduate students in the field of computer science (or related fields) and had an adequate educational background. To compare the two knowledge representation paradigms regarding their reasoning efficiency, both paradigms have been evaluated in two agent navigation scenarios of increasing size and complexity using the state-of-the-art solver CLINGO for ASP and an implementation of the reasoning algorithm from (Apeldoorn and Kern-Isberner 2016) for the more recent HKB approach.

In case of the comprehensibility, according to our study, participants made less mistakes using HKBs than using the ASP approach. However, the measured effect is rather small due to the overall small number of mistakes made by the participants. As a clear result, we showed that by far most of the participants prefer HKBs over ASP as a representation paradigm for the knowledge of agents (figure 4) which also conforms to the results of the preliminary study mentioned earlier (cf. footnote 6). Although HKBs were reported to have some disadvantages like there is no possibility to represent nescience and explicit negations, participants appreciated in general that HKBs are compact, clear and well structured.

In case of the reasoning efficiency, according to Figure 5 and Figure 6, we were able to show clearly that, in the context of our agent scenarios, reasoning through HKBs is much more efficient than through ASP using a state-of-the-art ASP solver.

According to our results, we can conclude that representing knowledge in the context of agents using HKBs seems to be more comprehensible. Moreover, HKBs offer much more efficient reasoning capabilities which are able to outperform a state-of-the-art ASP solver by far.

As future work, it could be of interest to compare the efficiency of the HKB reasoning algorithm to *minimal* answer set programs (instead of the answer set program automatically created from HKBs, cf. Section 4.2). Even if we do not expect this to close the huge efficiency gap between ASP and HKBs shown in our experiments, this would at least contribute to the completeness of our study.

Furthermore, since most of the participants were students of computer science or related fields (which obviously seems to be the main target audience for knowledge representation paradigms), it could also be of interest to have a closer look on broader groups of participants from other fields (the preliminary study mentioned in footnote 6 indicated already that there could be possible differences). Especially, this could be of interest, since in many scientific fields, students learn to think logically and in a structured way, which is an ability needed to solve and to understand paradigms like ASP, HKBs and other knowledge representation approaches.

An additional interesting point could be a further study which explores the comprehensibility of ASP an HKBs when rules are represented in form of (semi-)natural language. This would abstract from the formal representation of the two approaches.[9]

Finally, an interesting future task would be the transformation of answer set programs to HKBs (in (Apeldoorn and Kern-Isberner 2017) it is already outlined how HKBs can be transformed to answer set programs). By this, one could benefit both from the intuitive hierarchical representation of the knowledge and the faster reasoning capability of HKBs.

References

Apeldoorn, D., and Kern-Isberner, G. 2016. When should learning agents switch to explicit knowledge? In Benzmüller, C.; Sutcliffe, G.; and Rojas, R., eds., *GCAI 2016. 2nd Global Conference on Artificial Intelligence*, volume 41 of *EPiC Series in Computing*, 174–186. EasyChair Publications.

Apeldoorn, D., and Kern-Isberner, G. 2017. Towards an understanding of what is learned: Extracting multi-abstraction-level knowledge from learning agents. In Rus, V., and Markov, Z., eds., *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference*, 764–767. Palo Alto, CA, USA: AAAI Press.

Borgida, A., and Etherington, D. W. 1989. Hierarchical knowledge bases and efficient disjunctive reasoning. In Brachman, R. J.; Levesque, H. J.; and Reiter, R., eds., *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 33–43. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Kern-Isberner, G., and Beierle, C. 2014. *Methoden wissensbasierter Systeme – Grundlagen, Algortihmen, Anwendungen*. Wiesbaden: Springer Vieweg.

Krüger, C. 2016. *Statistische Evaluation unterschiedlicher Repräsentationsformen für die Wissensextraktion aus Reinforcement Learning*. Dortmund: Technische Universiät Dortmund.

Lang, J. 2015. *Twenty-Five Years of Preferred Subtheories*. Cham: Springer International Publishing. 157–172.

Zimbardo, P. G., and Gerrig, R. J. 2004. *Psychologie*. München: Pearson Studium.[10]

[9]Thanks to an anonymous reviewer for this idea.

[10]Thanks to Viola Gauß for pointing to this reference.