TECHNISCHE
UNIVERSITÄT
DRESDEN

Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Verification of Knowledge-Based Programs over Description Logic Actions

Benjamin Zarrieß Jens Claßen

LTCS-Report 15-10

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Verification of Knowledge-Based Programs over Description Logic Actions *†

Benjamin Zarrieß
Theoretical Computer Science
TU Dresden, Germany
zarriess@tcs.inf.tu-dresden.de

Jens Claßen
Knowledge-Based Systems Group
RWTH Aachen University, Germany
classen@kbsg.rwth-aachen.de

June 18, 2015

Abstract

A knowledge-based program defines the behavior of an agent by combining primitive actions, programming constructs and test conditions that make explicit reference to the agent's knowledge. In this paper we consider a setting where an agent is equipped with a Description Logic (DL) knowledge base providing general domain knowledge and an incomplete description of the initial situation. We introduce a corresponding new DL-based action language that allows for representing both physical and sensing actions, and that we then use to build knowledge-based programs with test conditions expressed in the epistemic DL. After proving undecidability for the general case, we then discuss a restricted fragment where verification becomes decidable. The provided proof is constructive and comes with an upper bound on the procedure's complexity.

*Supported by DFG Research Unit FOR 1513, project A1, http://www.hybrid-reasoning.org

†This is an extended version of an article in Proceedings of IJCAI-15 [ZC15]

Contents

1 Introduction

Since the Golog [LRL+97, DLL00] family of action programming languages has become a popular means for control of high-level agents, the verification of temporal properties of Golog programs has recently received increasing attention [CL08, DLP10]. Both the Golog language itself and the underlying Situation Calculus [MH69, Rei01a] are of high (first-order) expressivity, which renders the general problem undecidable. Identifying non-trivial fragments where decidability is given is therefore a worthwhile endeavour [DLP12, ZC14].

Here we consider the class of so-called *knowledge-based programs*, which are suited for more realistic scenarios where the agent possesses only incomplete information about its surroundings and has to use sensing in order to acquire additional knowledge at run-time. As opposed to classical Golog, knowledge-based programs contain explicit references to the agent's knowledge, thus enabling it to choose its course of action based on what it knows and does not know. Formalizations of knowledge-based programs were proposed by Reiter [Rei01b] and later by Claßen and Lakemeyer [CL06] based on Scherl and Levesque's [SL03] account of an epistemic Situation Calculus and Lakemeyer and Levesque's [LL04, LL10] modal variant $\mathcal{ES}$, respectively. Common to these approaches is that conditions in the program are evaluated by reducing reasoning about both knowledge and action to standard first-order theorem proving.

In this paper, we propose a new epistemic action formalism based on the basic Description Logic (DL) $\mathcal{ALC}$ by combining and extending earlier proposals for DL action formalisms [BLM+05] and epistemic DLs [DLN+98]. From the latter we use a concept constructor for knowledge to formulate test conditions within programs and desired properties thereof, while we extend the former by not only including physical, but also sensing actions. As will become apparent, representing and verifying knowledge-based programs with this language yields multiple advantages. First, we obtain decidability of verification for a formalism whose expressiveness goes far beyond propositional logic. Moreover, it enables us to resort to powerful DL reasoning systems [TH06, SLG14]. Finally, the new formalism also inherits many useful properties of the epistemic Situation Calculus and $\mathcal{ES}$ such as Reiter's [Rei91] solution to the frame problem, a variant of Levesque's [Lev90] notion of only-knowing, and a reasoning mechanism resembling Levesque and Lakemeyer's [LL01] Representation Theorem where reasoning about knowledge is reduced to standard DL reasoning.

As a motivating example, consider a mobile robot in a factory whose task it is to detect faults in gears and do the necessary repairs before turning them on. The agent is equipped with an (objective) DL knowledge base consisting of a TBox and an ABox, as usual. The TBox defines basic terminology such as the role name *has-f* that relates a system to its faults. The ABox gives an incomplete description of the initial situation and provides some properties of possible faults. Assume that the agent has two pure sensing actions at its disposal, namely $\texttt{sense-f}(\mathit{gear}, x)$ to sense whether the individual *gear* has fault x and $\texttt{sense-on}(\mathit{gear})$ to check if *gear* is on or not. Furthermore, the physical action $\texttt{repair}(\mathit{gear}, x)$ is available to remove a fault. An example for a knowledge-based program for this agent is given in Figure 1. As long as the agent does not know that *gear* has no known fault, a known fault x is chosen non-deterministically for which it is unknown whether *gear* has it or not. The agent then senses whether *gear* has this fault and repairs it if necessary. After completing the loop the agent turns on the gear system and checks if this was successful. If not, then there must be an unknown fault and an alarm is raised. An example for a property of this program to be verified is if a gear initially has an unknown critical fault, then the agent will eventually come to know it.

The remainder of this paper starts with recalling basic notion of DLs for representing initial knowledge, effect conditions of primitive actions, sensing properties, tests in programs and temporal properties of programs. In Section 3 we present our new action formalism that allows us to model both sensing and acting and consider the projection problem as a basic reasoning

```
while $\neg\mathsf{K}(\forall \mathit{has\text{-}f}.\neg\mathsf{K}\mathit{Fault})(\mathit{gear})$
  pick$(x)$ : $\mathsf{K}\mathit{Fault}(x) \land \neg\mathsf{K}\mathit{has\text{-}f}(\mathit{gear},x) \land \neg\mathsf{K}\neg\mathit{has\text{-}f}(\mathit{gear},x)$?.
    sense-f$(\mathit{gear},x)$;
    if $\mathsf{K}\mathit{has\text{-}f}(\mathit{gear},x)$ then repair$(\mathit{gear},x)$ else continue;
end
turn-on$(\mathit{gear})$; sense-on$(\mathit{gear})$;
if $\mathsf{K}\neg\mathit{On}(\mathit{gear})$ then raise-alarm else continue;
```

Figure 1: Example program

problem. Section 4 then is about the verification of programs. We define syntax and semantics of the programming language and show how to specify temporal properties of programs. Afterwards, we discuss a restricted fragment where verification becomes decidable and that in some respect even goes beyond earlier work on non-epistemic programs [ZC14], namely by re-introducing a limited variant of the operator for the non-deterministic choice of arguments ("pick operator"). We provide a constructive proof (along with an upper bound on the procedure's complexity) in which our variant of the Representation Theorem is used to build a finite abstraction of a program's transition system by means of DL reasoning, after which standard propositional model checking can be applied.

2 Epistemic Description Logic $\mathcal{ALCOK}$

The epistemic DL $\mathcal{ALCOK}$ extends the basic DL $\mathcal{ALC}$ by nominals ($\mathcal{O}$) i.e., singleton concepts and by a concept constructor for explicit references to knowledge ($\mathcal{K}$). In this section we recall the definitions of syntax and semantics of $\mathcal{ALCOK}$ following [DLN$^+$92, DLN$^+$98]. Although we often only use the objective sub-logics $\mathcal{ALC}$ and $\mathcal{ALCO}$ in this paper, we present all basic notions here for full $\mathcal{ALCOK}$.

Let N_R, N_C, N_I be countably infinite sets of *role names*, *concept names* and *individual names*, respectively.

A *generalized* $\mathcal{ALCOK}$*-role* P is built from role names using the *epistemic role constructor* and *role negation* (see Table 1). A *simple* $\mathcal{ALCOK}$*-role* (*role* for short) is a generalized role without role negation. $\mathcal{ALCOK}$*-concept descriptions* (*concepts* for short) are built from simple roles, concept names and the concept constructors shown in the lower part of Table 1.

In the following, we often use the symbols A, B to denote concept names, r for a role name, R for a role, P for a generalized role, a, b for individual names (*individuals* for short), and C, D, E, F, G for possibly complex concepts. In addition, we use $\mathsf{Kw}C$ as an abbreviation for $\mathsf{K}C \sqcup \mathsf{K}\neg C$. Intuitively, an object is an instance of $\mathsf{Kw}C$ if it is *known whether* it belongs to C or to $\neg C$.

The different kinds of $\mathcal{ALCOK}$-axioms are shown in Table 2. A TBox $\mathcal{T}$ is a finite set of *concept inclusions* (*CIs* for short) and an ABox is a finite set of *concept and role assertions* (also called *ABox assertions*) and *equality and inequality assertions*. A *concept definition* of the form $A \equiv C$ is viewed as an abbreviation of the two CIs $A \sqsubseteq C$ and $C \sqsubseteq A$. ABox assertions of the form $A(a), \neg A(a), r(a, b), \neg r(a, b)$ are called *literals*. A *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

The semantics of $\mathcal{ALCOK}$ is a *possible world semantics* defined in terms of interpretations.

	Syntax	*Semantics under* $(\mathcal{I}, \mathcal{W})$
role name	r	$r^{\mathcal{I}}$
role negation	$\neg P$	$(\Delta \times \Delta) \setminus P^{\mathcal{I},\mathcal{W}}$
epistemic role	$\mathsf{\mathbf{K}}P$	$\bigcap_{\mathcal{J}\in\mathcal{W}} P^{\mathcal{J},\mathcal{W}}$
concept name	A	$A^{\mathcal{I}}$
top	$\top$	Δ
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta \setminus C^{\mathcal{I},\mathcal{W}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I},\mathcal{W}} \cap D^{\mathcal{I},\mathcal{W}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I},\mathcal{W}} \cup D^{\mathcal{I},\mathcal{W}}$
existential restriction	$\exists R.C$	$\{d \mid \exists e : (d,e) \in R^{\mathcal{I},\mathcal{W}}, e \in C^{\mathcal{I},\mathcal{W}}\}$
value restriction	$\forall R.C$	$\{d \mid (d,e) \in R^{\mathcal{I},\mathcal{W}} \text{ implies } e \in C^{\mathcal{I},\mathcal{W}}\}$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
epistemic concept	$\mathsf{\mathbf{K}}C$	$\bigcap_{\mathcal{J}\in\mathcal{W}}(C^{\mathcal{J},\mathcal{W}})$

Table 1: Syntax and semantics of roles and concepts

		Axiom ϱ	$(\mathcal{I}, \mathcal{W}) \mid\!\models \varrho$, iff
TBox $\mathcal{T}$	concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I},\mathcal{W}} \subseteq D^{\mathcal{I},\mathcal{W}}$
	concept definition	$A \equiv C$	$A^{\mathcal{I},\mathcal{W}} = C^{\mathcal{I},\mathcal{W}}$
ABox $\mathcal{A}$	concept assertion	$C(a)$	$a \in C^{\mathcal{I},\mathcal{W}}$
	role assertion	$P(a,b)$	$(a,b) \in P^{\mathcal{I},\mathcal{W}}$
	equality assertion	$a \approx b$	$a = b$
	inequality assertion	$a \not\approx b$	$a \neq b$

Table 2: Syntax and semantics of axioms

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$ with $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for all $A \in N_C$, $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for all $r \in N_R$ and $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ for all $a \in N_I$. Here we adopt the so called *standard name assumption* (SNA): all interpretations are defined over a fixed countably infinite domain of standard names, denoted by Δ, and the interpretation of individuals is also fixed. W.l.o.g. we define $\Delta := N_I$. If not stated otherwise, we assume from now on that for any interpretation $\mathcal{I}$ it holds that $\Delta^{\mathcal{I}} = \Delta$ and $a^{\mathcal{I}} = a$ for all $a \in N_I$. An *epistemic interpretation* is a pair $(\mathcal{I}, \mathcal{W})$ consisting of an interpretation $\mathcal{I}$ and a set of interpretations $\mathcal{W}$. The interpretation function $\cdot^{\mathcal{I},\mathcal{W}}$ maps concepts and roles to subsets of Δ and $\Delta \times \Delta$, respectively, as given in Table 1. *Satisfaction of an axiom* ϱ in $(\mathcal{I}, \mathcal{W})$, denoted by $(\mathcal{I}, \mathcal{W}) \mid\!\models \varrho$, is defined as given in Table 2. An ABox $\mathcal{A}$ and TBox $\mathcal{T}$ is satisfied in $(\mathcal{I}, \mathcal{W})$, written as $(\mathcal{I}, \mathcal{W}) \mid\!\models \mathcal{T}$ and $(\mathcal{I}, \mathcal{W}) \mid\!\models \mathcal{A}$, respectively, if for all $\varrho \in \mathcal{T}$ and all $\varrho' \in \mathcal{A}$ it holds that $(\mathcal{I}, \mathcal{W}) \mid\!\models \varrho$ and $(\mathcal{I}, \mathcal{W}) \mid\!\models \varrho'$, respectively. A KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is satisfied in $(\mathcal{I}, \mathcal{W})$, written as $(\mathcal{I}, \mathcal{W}) \mid\!\models \mathcal{K}$, if $(\mathcal{I}, \mathcal{W}) \mid\!\models \mathcal{T}$ and $(\mathcal{I}, \mathcal{W}) \mid\!\models \mathcal{A}$.

An *epistemic model* of a KB $\mathcal{K}$ is a non-empty set of interpretations $\mathcal{M}$ such that:

- for all $\mathcal{I} \in \mathcal{M}$ it holds that $(\mathcal{I}, \mathcal{M}) \mid\!\models \mathcal{K}$ and
- for all sets of interpretations $\mathcal{M}'$ with $\mathcal{M} \subsetneq \mathcal{M}'$ there exists $\mathcal{J} \in \mathcal{M}'$ such that $(\mathcal{J}, \mathcal{M}') \mid\!\not\models \mathcal{K}$.

An axiom ϱ is *epistemically entailed* by $\mathcal{K}$, written as $\mathcal{K} \mid\!\models \varrho$, iff for all epistemic models $\mathcal{M}$ of

$\mathcal{K}$ and for all $\mathcal{I} \in \mathcal{M}$ it holds that $(\mathcal{I}, \mathcal{M}) \mid\!\models \varrho$. $\mathcal{K}$ is called *consistent* if $\mathcal{K}$ has an epistemic model.

A concept, axiom, TBox, ABox or KB without any occurrences of K is called *objective*, and *subjective* if all occurring concept and role names occur within the scope of a K. Note that for example a concept composed only of nominals is both objective and subjective since in our semantics an individual name is mapped to the same domain element in each possible interpretation. And a concept such as $A \sqcap \mathsf{K}B$ is neither subjective nor objective.

Since it holds that $(\mathsf{KK}P)^{\mathcal{I},\mathcal{W}} = (\mathsf{K}P)^{\mathcal{I},\mathcal{W}}$ for any epistemic interpretation $(\mathcal{I}, \mathcal{W})$ and generalized role P, we assume w.l.o.g. in the following that any simple role as used in concepts is either of the form r or $\mathsf{K}r$. In case of a non-empty set of interpretations $\mathcal{W}$ it also holds that $(\mathsf{K}\neg\mathsf{K}P)^{\mathcal{I},\mathcal{W}} = (\neg\mathsf{K}P)^{\mathcal{I},\mathcal{W}}$ and $(\mathsf{K}\neg\mathsf{K}C)^{\mathcal{I},\mathcal{W}} = (\neg\mathsf{K}C)^{\mathcal{I},\mathcal{W}}$.

If we deal with a *subjective concept or role* X, then we sometimes write $X^{\mathcal{W}}$ instead of $X^{\mathcal{I},\mathcal{W}}$ to denote the extension of X under an epistemic interpretation $(\mathcal{I}, \mathcal{W})$. In case of an objective concept C we sometimes write $C^{\mathcal{I}}$ instead of $C^{\mathcal{I},\mathcal{W}}$ and $\mathcal{I} \models X$ instead of $(\mathcal{I}, \mathcal{W}) \mid\!\models X$ for an objective axiom, KB, TBox or ABox X. Likewise, for an objective KB $\mathcal{K}$ and objective axiom ϱ we write $\mathcal{K} \models \varrho$ instead of $\mathcal{K} \mid\!\models \varrho$. An interpretation $\mathcal{I}$ *is a model of an objective KB* $\mathcal{K}$, denoted by $\mathcal{I} \models \mathcal{K}$, if all axioms in $\mathcal{K}$ are satisfied in $\mathcal{I}$.

The sublanguages $\mathcal{ALCO}$, $\mathcal{ALCK}$ or $\mathcal{ALC}$ of $\mathcal{ALCOK}$ are obtained by dropping either the K-constructor or nominals ($\{a\}$) or both, respectively. If we want to emphasize that a concept, role, axiom or KB is formulated in a specific DL $\mathcal{L} \in \{\mathcal{ALCO}, \mathcal{ALCK}, \mathcal{ALC}, \mathcal{ALCOK}\}$, then we write $\mathcal{L}$-concept, $\mathcal{L}$-role, $\mathcal{L}$-axiom or $\mathcal{L}$-KB, respectively.

The set of all subconcepts of a concept C is denoted by $\mathsf{sub}(C)$. For a given axiom, TBox, ABox or KB X the set $\mathsf{sub}(X)$ is defined as the set of all subconcepts of concepts occurring in X.

Usually, the standard semantics of non-epistemic DLs [BCM+03] is defined without the SNA, but w.r.t. first-order interpretations with an arbitrary (possibly finite) non-empty domain and an arbitrary interpretation of individual names. The consequences of the additional restrictions imposed by the SNA are for example discussed in detail in [MR11]. It holds that the semantics with SNA and the standard semantics are incompatible in the non-epistemic case. Clearly, there is no epistemic interpretation in which the objective CI $\top \sqsubseteq \{a\}$ is satisfied. However under the standard semantics $\top \sqsubseteq \{a\}$ is satisfiable.

Compatibility can be achieved by disallowing nominals in the knowledge base. This was shown in [MR11] for the DL $\mathcal{SRIQ}$ without the universal role. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an objective KB without nominals, ϱ an objective axiom and Ind a finite set of individuals that contains all individuals occurring in $\mathcal{K}$ and in ϱ. It holds that $\mathcal{K}' = (\mathcal{T}, \mathcal{A} \cup \{a \not\approx b \mid a, b \in \mathsf{Ind}\})$ has a model (under the standard semantics) iff $\mathcal{K}$ has an epistemic model and $\mathcal{K}' \models \varrho$ iff $\mathcal{K} \mid\!\models \varrho$. Intuitively, if $\mathcal{T}$ is nominal-free, then finiteness of the model cannot be enforced.

For an objective KB $\mathcal{K}$ there exists a unique epistemic model. This unique epistemic model of $\mathcal{K}$ is denoted by $\mathcal{M}(\mathcal{K})$. We have

$$\mathcal{M}(\mathcal{K}) = \{\mathcal{I} \mid \mathcal{I} \models \mathcal{K}, \Delta^{\mathcal{I}} = \Delta, a^{\mathcal{I}} = a \text{ for all } a \in N_I\}.$$

The main reasoning problem studied in [MR11,DLN+98] was deciding entailment of a subjective axiom w.r.t. an objective KB. Intuitively, the objective KB represents everything that is known about the world. The formal relationship to *only-knowing* [Lev90] is discussed in more detail in the next section.

3 Actions with Sensing

We introduce a simple notion of *primitive actions* describing the basic abilities of an agent to change the world and to gain new information from the environment. Primitive actions are defined as extra-logical syntactical objects composed of (parametrized) ABox assertions representing effects, effect conditions and properties of the environment that can be sensed, and are equipped with a purely model-based semantics. To justify the semantics of our action formalism we also present an embedding into a variant of the epistemic Situation Calculus $\mathcal{ES}$ [LL04, LL10]. Furthermore we investigate the complexity of projection.

3.1 Syntax and Semantics of Primitive Actions

To define the syntax of actions we first need some further preliminary notions:

Variables are taken from a countably infinite set N_V of variable names.

An *atom* is an ABox assertion where in place of individuals also variables are allowed, i.e. atoms are of the form $C(z)$, $P(z, z')$, $z \not\approx z'$ or $z \approx z'$, where $z, z' \in N_V \cup N_I$. *Primitive atoms* are of the form $A(z)$, $\neg A(z)$, $r(z, z')$ or $\neg r(z, z')$. A *formula* is a boolean combination of atoms. The set of all variables occurring in a formula ψ is denoted by $\mathsf{Var}(\psi)$. A formula without variables is called *ground formula*, i.e. it is a boolean combination of ABox assertions. Our notion of subjectivity and objectivity is extended to formulas in the obvious way. In the following we use the (possibly indexed) symbols φ for atoms, γ for primitive atoms and ψ for formulas.

Now we are ready to define the syntax of primitive actions.

Definition 1 (Primitive action)**.** An *effect* is either of the form ψ/γ (*conditional effect*) or of the form γ (*unconditional effect*), where ψ is an objective formula called *effect condition* and γ a primitive atom.

A *primitive action* α is a pair $\alpha = (\mathsf{eff}, \mathsf{sense})$ where eff is a set of effects and sense a set of objective formulas. The set of variables occurring in α is denoted by $\mathsf{Var}(\alpha)$. We often write $\alpha(x_1, \ldots, x_n)$ instead of just α where $\mathsf{Var}(\alpha) = \{x_1, \ldots, x_n\}$ are called *arguments of* α. A primitive action that contains no variables is called *ground action.* ▲

$$
\begin{aligned}
&\texttt{turn-on}(x) = (\mathsf{eff} : \{(\neg\exists \mathit{has\text{-}f}.\mathit{CritFault}(x))/\mathit{On}(x)\}, \mathsf{sense} : \emptyset)\\
&\texttt{repair}(x, y) = (\mathsf{eff} : \{\mathit{has\text{-}f}(x, y)/\neg \mathit{has\text{-}f}(x, y)\}, \mathsf{sense} : \emptyset)\\
&\texttt{sense-f}(x, y) = (\mathsf{eff} : \emptyset, \mathsf{sense} : \{\mathit{has\text{-}f}(x, y)\})\\
&\texttt{sense-on}(x) = (\mathsf{eff} : \emptyset, \mathsf{sense} : \{\mathit{On}(x)\})
\end{aligned}
$$

Figure 2: Example primitive actions

Figure 2 shows examples of two pure physical actions and two pure sensing actions. For instance consider the action $\texttt{turn-on}(x)$. It has a single conditional effect that causes x to be *On* after the action is executed only if x previously has no critical fault. Note that here effects and their conditions are restricted to be objective since eff is supposed to only encode physical effects. $\texttt{sense-on}(x)$ is a sensing action that represents the agent's ability to perceive whether $\mathit{On}(x)$ is true in the real world. Again, formulas in sense are objective since sensors only provide information about the outside world.

Semantically, a primitive action induces a binary relation on epistemic interpretations $(\mathcal{I}, \mathcal{W})$ which allows us to explicitly distinguish changes affecting the real world represented by $\mathcal{I}$ and changes to the knowledge state $\mathcal{W}$.

To execute an action we first need to instantiate it. A *variable mapping* ν is a total function of the form $\nu : N_V \to \Delta$. The formula $\psi^{x_1\cdots x_n}_{a_1\cdots a_n}$ where $x_i \in N_V$ and $a_i \in \Delta$ with $i = 1, \ldots, n$ and $n \geq 0$, denotes the formula that is obtained from ψ by simultaneously replacing each occurence of x_i in ψ by a_i for all $i = 1, \ldots, n$. Let ψ be a formula with $\mathsf{Var}(\psi) = \{x_1, \ldots, x_n\}$ and ν a variable mapping. We write ψ^ν as an abbreviation of the ground formula $\psi^{x_1\cdots x_n}_{\nu(x_1)\cdots\nu(x_n)}$.

Satisfaction of a ground formula in an epistemic interpretation and of an objective ground formula in a single interpretation is defined in the obvious way. Let P be a set of formulas and E a set of effects. We also use the following abbreviations defined by $\mathsf{P}^\nu := \{\psi^\nu \mid \psi \in \mathsf{P}\}$, $\neg\mathsf{P} := \{\neg\psi \mid \psi \in \mathsf{P}\}$ and $\mathsf{E}^\nu := \{\psi^\nu/\gamma^\nu \mid \psi/\gamma \in \mathsf{E}\} \cup \{\gamma^\nu \mid \gamma \in \mathsf{E}\}$. The instantiation of a primitive action $\alpha = (\mathsf{eff}, \mathsf{sense})$ using ν is defined by $\alpha^\nu = (\mathsf{eff}^\nu, \mathsf{sense}^\nu)$.

Next we define how a single interpretation is affected by a set unconditional ground effects.

Let $\mathcal{I}$ be an interpretation and L a set of unconditional ground effects, i.e. L is a set of literals. The *update* of $\mathcal{I}$ with L is an interpretation $\mathcal{I}^{\mathsf{L}}$, that is defined as follows:

- $A^{\mathcal{I}^{\mathsf{L}}} := (A^{\mathcal{I}} \setminus \{a \mid \neg A(a) \in \mathsf{L}\}) \cup \{a \mid A(a) \in \mathsf{L}\}$ for all $A \in N_C$ and
- $r^{\mathcal{I}^{\mathsf{L}}} := (r^{\mathcal{I}} \setminus \{(a, b) \mid \neg r(a, b) \in \mathsf{L}\}) \cup \{(a, b) \mid r(a, b) \in \mathsf{L}\}$ for all $r \in N_R$.

The following property of iterative updates of an interpretation is a direct consequence of the definition.

Lemma 2. *Let $\mathcal{I}$ be an interpretation and L_0 and L_1 two sets of literals. It holds that $(\mathcal{I}^{\mathsf{L}_0})^{\mathsf{L}_1} = \mathcal{I}^{\mathsf{L}'}$ with $\mathsf{L}' = \mathsf{L}_0 \setminus \neg\mathsf{L}_1 \cup \mathsf{L}_1$.*

For a given set of (possibly conditional) ground effects E and an interpretation $\mathcal{I}$ we define an *effect function* $\mathcal{E}(\cdot, \cdot)$ that maps E and $\mathcal{I}$ to a set of literals given by:

$$\mathcal{E}(\mathsf{E}, \mathcal{I}) := \{\gamma \mid \psi/\gamma \in \mathsf{E}, \mathcal{I} \models \psi\}.$$

To define how sensing affects the knowledge of the agent we adapt the notion of *sensing compatibility of worlds* from [LL04]:

Let $\alpha = (\mathsf{eff}, \mathsf{sense})$ be a primitive ground action and $\mathcal{I}$ and $\mathcal{J}$ two interpretations. $\mathcal{I}$ and $\mathcal{J}$ are sensing compatible w.r.t. α, written as $\mathcal{I} \sim_\alpha \mathcal{J}$, iff for all $\psi \in \mathsf{sense}$ it holds that $\mathcal{I} \models \psi$ iff $\mathcal{J} \models \psi$.

Now we are ready to define the execution semantics of a primitive ground action.

Definition 3. Let $\alpha = (\mathsf{eff}, \mathsf{sense})$ be a primitive ground action, $(\mathcal{I}, \mathcal{W})$ an epistemic interpretation with $\mathcal{I} \in \mathcal{W}$ and $(\mathcal{I}', \mathcal{W}')$ an epistemic interpretation. We write $(\mathcal{I}, \mathcal{W}) \Longrightarrow_\alpha (\mathcal{I}', \mathcal{W}')$, iff the following conditions are satisfied:

- $\mathcal{I}' = \mathcal{I}^{\mathsf{L}}$ with $\mathsf{L} = \mathcal{E}(\mathsf{eff}, \mathcal{I})$ and
- $\mathcal{W}' = \{\mathcal{J}^{\mathsf{L}} \mid \mathcal{J} \in \mathcal{W}, \mathcal{J} \sim_\alpha \mathcal{I}, \mathsf{L} = \mathcal{E}(\mathsf{eff}, \mathcal{J})\}$.

Let $\sigma = \beta_0, \ldots, \beta_{n-1}$ be a sequence of primitive ground actions. We write $(\mathcal{I}_0, \mathcal{W}_0) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$ as an abbreviation for $(\mathcal{I}_0, \mathcal{W}_0) \Longrightarrow_{\beta_0} (\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\beta_1} \cdots \Longrightarrow_{\beta_{n-1}} (\mathcal{I}_n, \mathcal{W}_n)$. ▲

The real world $\mathcal{I}$ of an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ is updated according to the physical effects eff. Intuitively, from $\mathcal{I}$ the agent receives information about the truth of each formula in sense. Interpretations in $\mathcal{W}$ contradicting this information are discarded, while those that agree with $\mathcal{I}$ are updated as well, yielding the new knowledge state. Thus, in our semantics the agent is fully aware of all effects of an action.

Note that it is ensured that $\mathcal{I} \in \mathcal{W}$ and $(\mathcal{I}, \mathcal{W}) \Longrightarrow_\alpha (\mathcal{I}', \mathcal{W}')$ implies $\mathcal{I}' \in \mathcal{W}'$.

3.2 Projection and the Relation to the Epistemic Situation Calculus

The projection problem is the problem to decide whether a given axiom holds after a sequence of actions has been performed. In our setting we assume that the knowledge of the agent about the initial world is represented as an $\mathcal{ALC}$-KB. Thus the agent has only incomplete knowledge about the world. Since the KB is objective, there exists according to the semantics only one unique epistemic model. This means we have complete information about what is known and what is not known about the world. In addition the agent has complete knowledge about how the primitive actions affect the state of the world. The *projection problem* is defined as follows.

Definition 4 (projection)**.** Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an objective KB, σ a sequence of primitive ground actions and Φ an $\mathcal{ALCOK}$-ground formula or $\mathcal{ALC}$-CI called *projection query*. We say that Φ is *valid after executing σ in $\mathcal{K}$* iff for all $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ it holds that $(\mathcal{I}', \mathcal{W}') \mid\!\models \Phi$ where $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}', \mathcal{W}')$. And we say that Φ is *satisfiable after executing σ in $\mathcal{K}$* iff there exists an $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ such that for the epistemic interpretation $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}', \mathcal{W}')$ it holds that $(\mathcal{I}', \mathcal{W}') \mid\!\models \Phi$. ▲

It clearly holds that Φ is valid after executing σ in $\mathcal{K}$ iff $\neg\Phi$ is not satisfiable after executing σ in $\mathcal{K}$. Let Ind be the set of individuals occurring in $\mathcal{K}$ and σ and $c_u \in \Delta \setminus \mathsf{Ind}$. It holds that the $\mathcal{ALC}$-CI $C \sqsubseteq D$ is valid after executing σ in $\mathcal{K}$ iff the $\mathcal{ALC}$-ground formula given by

$$\bigvee_{a \in \mathsf{Ind} \cup \{c_u\}} (C \sqcap \neg D)(a)$$

is not satisfiable after executing σ in $\mathcal{K}$. In the following we focus only on the validity problem of $\mathcal{ALCOK}$-ground formulas as projection queries. All other variants of the projection problem can be reduced to this case.

Note that the TBox $\mathcal{T}$ is only required to hold and to be known in the initial state, and that later states resulting from the execution of actions may violate it. While the persistence of $\mathcal{T}$ is thus not enforced in our formalization, checking this property is simply a special case of the projection problem.

$$\begin{aligned}\mathcal{T} = \{&\mathit{Fault} \sqsubseteq \mathit{CritFault} \sqcup \mathit{UncritFault},\\ &\exists \mathit{has\text{-}f}.\top \sqsubseteq \mathit{System}, \mathit{System} \sqsubseteq \forall \mathit{has\text{-}f}.\mathit{Fault}\}\\ \mathcal{A} = \{&\mathit{System}(\mathit{gear}), \neg \mathit{On}(\mathit{gear}), \mathit{Fault}(\mathit{blocked})\}\end{aligned}$$

Figure 3: Example initial knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$

Example 5. Figure 3 shows an initial KB $\mathcal{K}$ for our example domain. The first CI in $\mathcal{T}$ states that faults are critical faults or uncritical ones, the last two CIs define the domain *System* and range *Fault* for the role *has-f*. $\mathcal{A}$ describes a simple initial situation. Assume $\mathcal{K}$ is all the agent knows initially about the world. Thus, it is known that *gear* is *not on*, but the effect condition $\neg\exists \mathit{has\text{-}f}.\mathit{CritFault}(\mathit{gear})$ of $\mathtt{turn\text{-}on}(\mathit{gear})$ is unknown (there is a least one possible world in $\mathcal{M}(\mathcal{K})$ satisfying it and one that does not). Consequently, after executing $\mathtt{turn\text{-}on}(\mathit{gear})$ in $\mathcal{K}$, we get that $\neg\mathsf{\mathbf{Kw}}\, \mathit{On}(\mathit{gear})$ is valid. If the agent now in turn executes $\mathtt{sense\text{-}on}(\mathit{gear})$, it will also come to know whether *gear* has a critical fault, i.e. both $\mathbf{Kw}\exists \mathit{has\text{-}f}.\mathit{CritFault}(\mathit{gear})$ and $\mathbf{Kw}\,\mathit{On}(\mathit{gear})$ are valid. ▲

In the following we show that the projection problem can be equivalently formulated as an entailment problem in the first-order modal logic $\mathcal{ES}$ [LL04, LL10], that was designed for reasoning about knowledge and action and is capable of representing Reiter-style *basic action theories* (BAT) extended with an account of sensing.

A first epistemic extension of the classical situation calculus [Rei01a, MH69] was proposed by Scherl and Levesque [SL03] . However in Scherl and Levesque's purely axiomatic formalization complete information about knowledge cannot be represented. An explicit axiomatization of what is known and also of what is not known would be required to achieve this. In $\mathcal{ES}$ there is a a modal operator for only-knowing to overcome this problem. Furthermore, in $\mathcal{ES}$ we also have a fixed countably infinite domain and an embedding of the introduced DL-based action formalism into $\mathcal{ES}$ is rather straightforward.

First, we recall the main definitions of the logic $\mathcal{ES}$ and *basic action theories* (BATs) according to [LL04, Cla14].

We define a set of *terms* as follows.

Definition 6 (terms)**.** There are terms of two sorts *object* and *action.* They can be built using the following symbols:

- variables $x, y, \cdots$ of sort *object*;
- a single variable a of sort *action*;
- a countably infinite set N_O of *rigid object constant symbols*;
- a non-empty set N_A of *rigid action function symbols* with arguments of sort object;

Every variable is a term, every $c \in N_O$ is a term and if α is an action function of arity k and $t_1, \ldots, t_k$ are terms of sort object, then also $\alpha(t_1, \ldots, t_k)$ is a term. A term is called *ground term* if it contains no variables. We denote the set of all ground terms (also called *standard names*) of sort object by $\mathcal{N}_O$ (i.e. $\mathcal{N}_O = N_O$), and those of sort action by $\mathcal{N}_A$. ▲

To build formulas we use predicate symbols (called *fluents*) taken from a set N_F with arguments of sort object and two distinguished unary predicates *Poss* and *SF*, each with one argument of sort action.

Formulas are then built using the usual logical connectives and in addition we have modal operators $[\cdot]$ and $\Box$ referring to future situations and *Know* and *OKnow* for knowledge.

Definition 7 ($\mathcal{ES}$-Formulas)**.** The set of formulas is defined as the least set satisfying the following conditions:

- If $t_1, ..., t_k$ are terms of sort object, t_a a term of sort action and $F \in N_F$ a k-ary fluent, then $F(t_1, ..., t_k)$, $\mathit{Poss}(t_a)$ and $\mathit{SF}(t_a)$ are formulas.
- If t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.
- If ϕ and ϕ' are formulas, v a variable and t a term of sort action, then $\phi \wedge \phi'$, $\neg\phi$, $\forall v.\phi$, $\Box\phi$, $[t]\phi$, $\mathit{Know}(\phi)$ and $\mathit{OKnow}(\phi)$ are formulas.

We understand $\vee$, $\exists$, $\supset$ and $\equiv$ as the usual abbreviations and use *true* for a tautology. A formula $[t_0][t_1]\cdots[t_n]\phi$ is abbreviated by $[t_0 t_1 \cdots t_n]\phi$. A formula is called *fluent formula* if it contains no $\Box$, no $[\cdot]$ and not the predicates *Poss* and *SF*. A *fluent sentence* is a fluent formula without free variables. ▲

Intuitively, the formula $[t]\phi$ means that ϕ is true after doing action t and $\Box\phi$ means that ϕ holds after any sequence of actions. The semantics of formulas is defined in terms of *worlds*.

Definition 8 (Worlds)**.** *Primitive formulas* are of the form $F(d_1, ..., d_k)$, $Poss(t_a)$, $SF(t_a)$ where $F \in N_F$ is k-ary fluent, $d_i \in \mathcal{N}_O$ for all $i = 1, \dots, k$ and $t_a \in \mathcal{N}_A$. The *set of all primitive formulas* is denoted by $\mathcal{P}_F$ and the *set of all action sequences* is given by $\mathcal{Z} := \mathcal{N}_A^*$. A *world* w is a total function of the form

$$w : \mathcal{P}_F \times \mathcal{Z} \to \{0, 1\}.$$

The set of all worlds is denoted by W. A set of worlds $e \subseteq W$ is called *epistemic state*. ▲

A world thus maps primitive formulas to truth values. To define the meaning of the epistemic operators *Know* and *OKnow* some additional notions are needed.

We use the symbol $\langle\rangle$ to denote the empty sequence of action standard names.

Let $w, w' \in W$, $\sigma \in \mathcal{Z}$ and $t \in \mathcal{N}_A$. Sensing compatibility of w and w' w.r.t. σ, denoted by $w \simeq_\sigma w'$, is defined inductively as follows: It holds that $w \simeq_{\langle\rangle} w'$. It holds that $w \simeq_{\sigma \cdot t} w'$ iff $w \simeq_\sigma w'$ and $w[SF(t), \sigma] = w'[SF(t), \sigma]$.

Let $w \in W$ and $\sigma \in \mathcal{Z}$. The *progression of w through σ* is a world w_σ such that $w_\sigma[\xi, \sigma'] = w[\xi, \sigma \cdot \sigma']$ for all $\xi \in \mathcal{P}_F$ and all $\sigma' \in \mathcal{Z}$.

Let e be an epistemic state, $w \in W$ and $\sigma \in \mathcal{Z}$. The progression of e through σ w.r.t. w, denoted by e^w_σ, is an epistemic state, that is defined as follows:

$$e^w_\sigma := \{w'_\sigma \mid w' \in e, w' \simeq_\sigma w\}.$$

We are now equipped to define the truth of formulas:

Definition 9 (Satisfaction of Formulas)**.** Given an epistemic state $e \subseteq W$, a world $w \in W$ and a sentence ϕ, we define $e, w \models \phi$ as $e, w, \langle\rangle \models \phi$, where for any $\sigma \in \mathcal{Z}$:

1. $e, w, \sigma \models \xi$ iff $w[\xi, \sigma] = 1$ for all $\xi \in \mathcal{P}_F$;
2. $e, w, \sigma \models (c_1 = c_2)$ iff c_1 and c_2 are identical;
3. $e, w, \sigma \models \phi_1 \wedge \phi_2$ iff $e, w, \sigma \models \phi_1$ and $e, w, \sigma \models \phi_2$;
4. $e, w, \sigma \models \neg\phi$ iff $e, w, \sigma \not\models \phi$;
5. $e, w, \sigma \models \forall v.\phi$ iff $e, w, \sigma \models \phi^v_d$ for all $d \in \mathcal{N}_v$;
6. $e, w, \sigma \models \Box\phi$ iff $e, w, \sigma \cdot \sigma' \models \phi$ for all $\sigma' \in \mathcal{Z}$;
7. $e, w, \sigma \models [t]\phi$ iff $e, w, \sigma \cdot t \models \phi$;
8. $e, w, \sigma \models Know(\phi)$ iff for all $w' \in e^w_\sigma$: $e^w_\sigma, w', \langle\rangle \models \phi$;
9. $e, w, \sigma \models OKnow(\phi)$ iff for all $w' \in W$: $w' \in e^w_\sigma$ iff $e^w_\sigma, w', \langle\rangle \models \phi$.

▲

Above, $\mathcal{N}_v$ refers to the set of all standard names of the same sort as the variable v. We moreover use the notation ϕ^v_t denoting the result of simultaneously replacing all free occurrences of v by the term t of the same sort as v in ϕ.

We recall the definition of a basic action theory.

Definition 10. A *basic action theory* (BAT)

$$\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{pre}} \cup \mathcal{D}_{\text{post}} \cup \mathcal{D}_{\text{sense}}$$

describes the dynamics of a specific application domain, where

1. $\mathcal{D}_0$, the *initial theory*, is a finite set of fluent sentences describing the initial state of the world.
2. $\mathcal{D}_{\text{pre}}$ is a set containing a single *precondition axiom* of the form

   $$\forall a.\Box\big(Poss(a) \equiv \vartheta\big)$$

   where ϑ is a fluent formula.
3. $\mathcal{D}_{\text{post}}$ is a finite set of *successor state axioms* (SSAs), one for each fluent relevant to the application domain, incorporating Reiter's [Rei01a] solution to the frame problem, and encoding the effects the actions have on the different fluents. The SSA for a fluent predicate has the form

   $$\forall a.\forall \vec{x}.\Box\big(\big([a]F(\vec{x})\big) \equiv \gamma^+_F \vee F(\vec{x}) \wedge \neg\gamma^-_F\big)$$

   where the *positive effect condition* γ^+_F and *negative effect condition* γ^-_F are fluent formulas with free variables $\vec{x}$ and a.
4. $\mathcal{D}_{\text{sense}}$ contains a single sentence of the following form

   $$\forall a.\Box\big(SF(a) \equiv \varsigma\big)$$

   where ς is a fluent formula.

▲

First, we define a translation of generalized $\mathcal{ALCOK}$-roles, $\mathcal{ALCOK}$-concepts, $\mathcal{ALCOK}$-axioms and $\mathcal{ALCOK}$-formulas to fluent formulas in $\mathcal{ES}$, that is basically defined as the well-known standard translation of DLs into first-order logic. In addition we replace K by the *Know*-constructor available in $\mathcal{ES}$. Concept names are identified with unary fluent predicates, role names with binary fluents and individual names from N_I with rigid constant symbols in N_O. We define $N_F := N_C \cup N_R$ and $N_O := N_I$.

Definition 11. Let x, y be variables of sort object. The function $\mathsf{tr}_{x,y}(\cdot)$ maps a generalized role to an $\mathcal{ES}$-fluent formula with exactly the free variables x and y. It is inductively defined by

$$\begin{aligned}
\mathsf{tr}_{x,y}(r) &:= r(x,y) \text{ with } r \in N_R;\\
\mathsf{tr}_{x,y}(\neg P) &:= \neg\mathsf{tr}_{x,y}(P);\\
\mathsf{tr}_{x,y}(\mathsf{K}P) &:= \mathit{Know}(\mathsf{tr}_{x,y}(P)).
\end{aligned}$$

The functions $\mathsf{tr}_x(\cdot)$ and $\mathsf{tr}_y(\cdot)$ map a given $\mathcal{ALCOK}$-concept C to a fluent formula $\mathsf{tr}_x(C)$ and $\mathsf{tr}_y(C)$ with exactly one free variable of sort object x and y, respectively. $\mathsf{tr}_x(\cdot)$ is inductively defined as follows:

$$\begin{aligned}
\mathsf{tr}_x(\top) &:= x = x\\
\mathsf{tr}_x(A) &:= A(x) \text{ with } A \in N_C\\
\mathsf{tr}_x(\{a\}) &:= x = a\\
\mathsf{tr}_x(\neg D) &:= \neg\mathsf{tr}_x(D)\\
\mathsf{tr}_x(D_1 \sqcap D_2) &:= \mathsf{tr}_x(D_1) \wedge \mathsf{tr}_x(D_2)\\
\mathsf{tr}_x(\exists R.D) &:= \exists y.\big(\mathsf{tr}_{x,y}(R) \wedge \mathsf{tr}_y(D)\big)\\
\mathsf{tr}_x(\mathbf{K}D) &:= \mathit{Know}(\mathsf{tr}_x(D))
\end{aligned}$$

The mapping $\mathsf{tr}_y(\cdot)$ is defined as $\mathsf{tr}_x(\cdot)$ but with x and y swapped. For $\mathcal{ALCOK}$-CIs and $\mathcal{ALCOK}$-atoms a translation $\mathsf{tr}(\cdot)$ is defined as follows, where variables from N_V are viewed as variables of sort object:

$$\begin{aligned}
\mathsf{tr}(C \sqsubseteq D) &:= \forall x.(\mathsf{tr}_x(C) \supset \mathsf{tr}_x(D))\\
\mathsf{tr}(C(z)) &:= \big(\mathsf{tr}_x(C)\big)^{x}_{z}\\
\mathsf{tr}(P(z,z')) &:= \big(\mathsf{tr}_{x,y}(P)\big)^{x\ y}_{z\ z'}\\
\mathsf{tr}(z \not\approx z') &:= z \neq z'.
\end{aligned}$$

The translation $\mathsf{tr}(\psi)$ of an $\mathcal{ALCOK}$-formula ψ is defined in the obvious way. ▲

To show that this translation is correct we define a compatibility relation "$\cong_{\mathcal{R}}$" between DL interpretations and $\mathcal{ES}$-worlds in the obvious way. Note that $\mathcal{N}_O = \Delta$.

Definition 12. Let $\mathcal{R} \subseteq N_C \cup N_R$ be a finite set of names, $\mathcal{I}$ an interpretation, w a world and $\sigma \in \mathcal{Z}$. We write $\mathcal{I} \cong_{\mathcal{R}} (w,\sigma)$ iff

- for all concept names $A \in \mathcal{R}$ and all $d \in \Delta$ it holds that $w[A(d),\sigma] = 1$ iff $d \in A^{\mathcal{I}}$ and
- for all role names $r \in \mathcal{R}$ and all $d, d' \in \Delta$ it holds that $w[r(d,d'),\sigma] = 1$ iff $(d,d') \in r^{\mathcal{I}}$.

We use $\mathcal{I} \cong_{\mathcal{R}} w$ as abbreviation for $\mathcal{I} \cong_{\mathcal{R}} (w, \langle\rangle)$. We extend this embedding to the epistemic case as follows: Let $e \subseteq W$ be an epistemic state, w a world and $(\mathcal{I},\mathcal{W})$ an epistemic interpretation. We write $(\mathcal{I},\mathcal{W}) \cong_{\mathcal{R}} (e,w,\sigma)$ iff

- $\mathcal{I} \cong_{\mathcal{R}} (w,\sigma)$ and
- for all $\mathcal{I}' \in \mathcal{W}$ there exists $w' \in e^w_\sigma$ such that $\mathcal{I}' \cong_{\mathcal{R}} w'$ and
- for all $w' \in e^w_\sigma$ there exists $\mathcal{I}' \in \mathcal{W}$ such that $\mathcal{I}' \cong_{\mathcal{R}} w'$.

▲

Basically, the two semantic structures are in "$\cong_{\mathcal{R}}$"-relation if both agree on the truth values of the atomic formulas that can be built from names in $\mathcal{R}$. Therefore, the following lemma is a direct consequence of the definition of $\cong_{\mathcal{R}}$ and the definition of the translation functions.

Lemma 13. *Let $(\mathcal{I},\mathcal{W})$ be an epistemic interpretation, $w \in W$ a world, $e \subseteq W$ a set of worlds, $\sigma \in \mathcal{Z}$ such that $(\mathcal{I},\mathcal{W}) \cong_{\mathcal{R}} (e,w,\sigma)$, P a generalized role C a concept and ψ an $\mathcal{ALCOK}$-axiom or ground formula with names only from $\mathcal{R}$. It holds that*

1. *$(d,d') \in P^{\mathcal{I},\mathcal{W}}$ iff $e,w,\sigma \models \big(\mathsf{tr}_{x,y}(P)\big)^{x\ y}_{d\ d'}$ for all $d,d' \in \Delta$;*

2. *$d \in C^{\mathcal{I},\mathcal{W}}$ iff $e, w, \sigma \models \big(\mathsf{tr}_x(C)\big)^x_d$ for all $d \in \Delta$ and*

3. *$(\mathcal{I},\mathcal{W}) \mid\models \psi$ iff $e, w, \sigma \models \mathsf{tr}(\psi)$.*

Proof. The first two claims can be shown by induction on the structure of P and C using the definition of the translation functions and the compatibility relation. The third claim is a consequence of the first two and the definition of the translation. We omit a detailed proof here. □

Now we are ready to axiomatize the meaning of primitive actions, defined in the previous section, as a BAT using the embedding of $\mathcal{ALCOK}$-axioms and formulas into fluent formulas in $\mathcal{ES}$. However we first need some minor additional restrictions on the primitive actions. In $\mathcal{ES}$ each ground action t_a has a single (binary) sensing result given by the truth value of $SF(t_a)$ whereas in the definition of primitive DL actions sense is a set of formulas and we therefore obtain a (possibly empty) set of binary sensing results. For the sake of simplicity we consider here only primitive actions $\alpha = (\mathsf{eff}, \mathsf{sense})$ where sense is a singleton set. In case of a purely physical α we simply assume that $\mathsf{sense} : \{\top(b)\}$ for some $b \in \Delta$. Furthermore, we assume that unconditional effects $\gamma \in \mathsf{eff}$ are written as conditional ones of the form $\top(b)/\gamma$. Note that also the general case with a set of sensing results can be easily modeled in $\mathcal{ES}$ by generalizing some of the definitions related to sensing. For example one could introduce a sufficient finite number of different sensing fluents, add a corresponding axiom to $\mathcal{D}_{\text{sense}}$ for each of them and generalize the definition of sensing compatibility accordingly.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALC}$-KB and $\Sigma = \{\alpha_1(\vec{x}^1), \dots, \alpha_n(\vec{x}^n)\}$ a finite set of primitive actions with $\alpha_i(\vec{x}^i) = (\mathsf{eff}_i, \mathsf{sense}_i : \{\varsigma_i\})$ and variables $\vec{x}^i = (x^i_1, \dots, x^i_{m_i})$ as arguments.

From now we consider a fixed finite *set of relevant names*, denoted by $\mathcal{R} \subset N_C \cup N_R$, that contains all concept and role names occurring in $\mathcal{K}$, Σ and in the projection query. For the definition of the BAT each primitive action $\alpha_i(\vec{x}^i) \in \Sigma$ is identified with an action function $\alpha_i \in N_A$ with arity m_i. The *set of all ground actions* obtained from Σ is given by

$$\Sigma_{\mathsf{g}} := \{\alpha^\nu \mid \alpha \in \Sigma, \nu \text{ is a variable mapping}\}.$$

For the translation we simply view an instantiated actions α^ν as a primitive ground action term in $\mathcal{ES}$ in the obvious way. Thus, we have $\Sigma_{\mathsf{g}} \subseteq \mathcal{N}_A$ and $\Sigma^*_{\mathsf{g}} \subseteq \mathcal{Z}$.

We construct the BAT $\mathcal{D}_{\mathcal{K},\Sigma} = \mathcal{D}^{\mathcal{K}}_0 \cup \mathcal{D}^{\Sigma}_{\text{pre}} \cup \mathcal{D}^{\Sigma}_{\text{post}} \cup \mathcal{D}^{\Sigma}_{\text{sense}}$ as follows: The KB $\mathcal{K}$ represents the initial theory:

$$\mathcal{D}^{\mathcal{K}}_0 = \{\mathsf{tr}(\varrho) \mid \varrho \in \mathcal{T} \cup \mathcal{A}\}.$$

The primitve actions don't have preconditions, i.e. they are always possible. Thus, we define $\mathcal{D}^{\Sigma}_{\text{pre}} = \{\forall a.\Box\big(Poss(a) \equiv \vartheta\big)\}$ with

$$\vartheta = \bigvee_{i \in \{1,\dots,n\}} \exists \vec{x}^i.\big(a = \alpha_i(\vec{x}^i)\big).$$

$\mathcal{D}^{\Sigma}_{\text{post}}$ consists of an SSA for each concept name A and role name r in $\mathcal{R}$ of the form

$$\forall a.\forall x.\Box\big([a]A(x) \equiv \gamma_A\big) \text{ and } \forall a.\forall x,y.\Box\big([a]r(x,y) \equiv \gamma_r\big), \text{ respectively,}$$

with $\gamma_A = \gamma^+_A \vee A(x) \wedge \neg\gamma^-_A$ and $\gamma_r = \gamma^+_r \vee r(x,y) \wedge \neg\gamma^-_r$. The positive and negative effect conditions γ_A^+ and γ_A^- are defined according to the sets eff_i.

$$\gamma_A^+ = \bigvee_{\substack{i\in\{1,\dots,n\}\\ \psi/A(z)\in\mathsf{eff}_i}} \exists\vec{x}^i.\big(a = \alpha_i(\vec{x}^i) \wedge x = z \wedge \mathsf{tr}(\psi)\big)$$
$$\gamma_A^- = \bigvee_{\substack{i\in\{1,\dots,n\}\\ \psi/\neg A(z)\in\mathsf{eff}_i}} \exists\vec{x}^i.\big(a = \alpha_i(\vec{x}^i) \wedge x = z \wedge \mathsf{tr}(\psi)\big).$$

The effect conditions for roles are defined following the same scheme as for concept names.

$$\gamma_r^{+(-)} = \bigvee_{\substack{i\in\{1,\dots,n\}\\ \psi/(\neg)r(z,z')\in\mathsf{eff}_i}} \exists\vec{x}^i.\big(a = \alpha_i(\vec{x}^i) \wedge x = z \wedge y = z' \wedge \mathsf{tr}(\psi)\big).$$

As for $\mathcal{D}_{\text{pre}}^{\Sigma}$, the set $\mathcal{D}_{\text{sense}}^{\Sigma}$ contains a single definition of the form $\forall a.\Box\big(SF(a) \equiv \gamma_{SF}\big)$ with

$$\gamma_{SF} = \bigvee_{i\in\{1,\dots,n\}} \exists\vec{x}^i.\big(a = \alpha_i(\vec{x}^i) \wedge \mathsf{tr}(\varsigma_i)\big).$$

The projection problem can be formulated as an entailment problem in $\mathcal{ES}$ as follows:

$$OKnow(\mathcal{D}) \wedge \mathcal{D}' \models [\sigma]\phi$$

where $\mathcal{D}$ is a BAT that defines everything that is known about the world, $\mathcal{D}'$ a second BAT describing the world as it is assumed to be in reality, $\sigma \in \mathcal{Z}$ an action sequence and the projection query ϕ is a fluent sentence.

We will show the following: An $\mathcal{ALCOK}$-ground formula or CI ψ is valid after executing the ground action sequence $\sigma \in \Sigma_{\mathsf{g}}^*$ in the KB $\mathcal{K}$ iff $OKnow(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma} \models [\sigma]\mathsf{tr}(\psi)$.

For the proof we need some further preliminaries.

Let $\mathcal{I}$ be an interpretation, w a world and $\mathcal{D}_{\mathcal{K},\Sigma}$ the BAT as constructed above. We define a world $w_{\mathcal{I}}$ that coincides initially with $\mathcal{I}$ and satisfies the SSAs, the precondition axiom and the sensing axiom in $\mathcal{D}_{\mathcal{K},\Sigma}$. $w_{\mathcal{I}}$ is a world satisfying the following conditions:

1. For all concept names $A \in \mathcal{R}$ and all $d \in \mathcal{N}_O$ it holds that:
   (a) $w_{\mathcal{I}}[A(d), \langle\rangle] = 1$ iff $d \in A^{\mathcal{I}}$ and
   (b) $w_{\mathcal{I}}[A(d), \sigma \cdot t] = 1$ iff $w_{\mathcal{I}}, \sigma \models (\gamma_A)^{x\ a}_{d\ t}$ for all $\sigma \cdot t \in \mathcal{Z}$.
2. For all role names $r \in \mathcal{R}$ and all $d, d' \in \mathcal{N}_O$ it holds that
   (a) $w_{\mathcal{I}}[r(d,d'), \langle\rangle] = 1$ iff $(d, d') \in r^{\mathcal{I}}$ and
   (b) $w_{\mathcal{I}}[r(d,d'), \sigma \cdot t] = 1$ iff $w_{\mathcal{I}}, \sigma \models (\gamma_r)^{x\ y\ a}_{d\ d'\ t}$ for all $\sigma \cdot t \in \mathcal{Z}$.
3. For all $F(d_1, \dots, d_k) \in \mathcal{P}_F$ with $F \in N_F$ and $F \notin \mathcal{R}$ and all $\sigma \in \mathcal{Z}$ it holds that $w_{\mathcal{I}}[F(d_1, \dots, d_k), \sigma] = w[F(d_1, \dots, d_k), \sigma]$.
4. For all $t \in \mathcal{N}_A$ and all $\sigma \in \mathcal{Z}$ it holds that
   (a) $w_{\mathcal{I}}[Poss(t), \sigma] = 1$ and
   (b) $w_{\mathcal{I}}[SF(t), \sigma] = 1$ iff $w_{\mathcal{I}}, \sigma \models (\gamma_{SF})^a_t$.

The definition of $w_{\mathcal{I}}$ is very similar to Definition 1 in [LL04] and as in [LL04] we can show the following properties:

Lemma 14. *Let $\mathcal{I}$ be an interpretation and w a world. It holds that:*

1. *$w_{\mathcal{I}}$ exists and is uniquely determined;*
2. *$\mathcal{I} \cong_{\mathcal{R}} w_{\mathcal{I}}$;*
3. *If $\mathcal{I} \in \mathcal{M}(\mathcal{K})$, then $w_{\mathcal{I}} \models \mathcal{D}_{\mathcal{K},\Sigma}$;*
4. *If $\mathcal{I} \cong_{\mathcal{R}} w$ and $w \models \mathcal{D}_{\mathcal{K},\Sigma}$, then $w = w_{\mathcal{I}}$.*

Proof. 1. Obviously, for given $\mathcal{I}$, w and $\mathcal{D}_{\mathcal{K},\Sigma}$ the world $w_{\mathcal{I}}$ exists and is uniquely determined.

2. $\mathcal{I} \cong_{\mathcal{R}} w_{\mathcal{I}}$ follows directly from the conditions 1. (a) and 2. (a).

3. Since $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $\mathcal{I} \cong_{\mathcal{R}} w_{\mathcal{I}}$ it follows with Lemma 13 that $w_{\mathcal{I}} \models \mathcal{D}_0^{\mathcal{K}}$. By definition $w_{\mathcal{I}}$ satisfies also the SSAs and the precondition and sensing axiom. Therefore, $w_{\mathcal{I}} \models \mathcal{D}_{\mathcal{K},\Sigma}$.

4. With $\mathcal{I} \cong_{\mathcal{R}} w$ it follows that w satisfies the conditions 1. (a) and 2. (a). And since $w \models \mathcal{D}_{\mathcal{K},\Sigma}$ also the remaining conditions are satisfied.

□

In the next lemma we show that the construction of $\mathcal{D}_{\mathcal{K},\Sigma}$ ensures that the relation $\cong_{\mathcal{R}}$ is preserved after executing an action.

Lemma 15. *Let $\mathcal{I}$ be an interpretation, $w \in W$ a world with $w \models \mathcal{D}_{\mathcal{K},\Sigma}$, $\sigma \in \mathcal{Z}$ an action sequence and $\beta = (\mathsf{eff}, \mathsf{sense}) \in \Sigma_{\mathsf{g}}$. It holds that $\mathcal{I} \cong_{\mathcal{R}} (w, \sigma)$ implies $\mathcal{I}^{\mathcal{E}(\mathsf{eff},\mathcal{I})} \cong_{\mathcal{R}} (w, \sigma \cdot \beta)$.*

Proof. Let $\mathcal{I}' = \mathcal{I}^{\mathcal{E}(\mathsf{eff},\mathcal{I})}$. We show that $\mathcal{I} \cong_{\mathcal{R}} (w, \sigma)$ implies $\mathcal{I}' \cong_{\mathcal{R}} (w, \sigma \cdot \beta)$ given that $w \models \mathcal{D}_{\mathcal{K},\Sigma}$. Let $A \in \mathcal{R}$ be a concept name.

We show that $w[A(d), \sigma \cdot \beta] = 1$ iff $d \in A^{\mathcal{I}'}$ for $d \in \Delta$.

$\Rightarrow$: First, assume $w[A(d), \sigma \cdot \beta] = 1$. Since $w \models \mathcal{D}_{\text{post}}^{\Sigma}$ it is implied by Lemma 14 that

$$w, \sigma \models \big(\gamma_A^+\big)^{x\ a}_{d\ \beta} \vee A(d) \wedge \neg\big(\gamma_A^-\big)^{x\ a}_{d\ \beta}.$$

First, assume $w, \sigma \models \big(\gamma_A^+\big)^{x\ a}_{d\ \beta}$, i.e. by definition of γ_A^+ there exists $\psi/A(d) \in \mathsf{eff}$ such that $w, \sigma \models \mathsf{tr}(\psi)$. Since $\mathcal{I} \cong_{\mathcal{R}} (w, \sigma)$ by assumption it follows that $\mathcal{I} \models \psi$ and therefore $A(d) \in \mathcal{E}(\mathsf{eff}, \mathcal{I})$. With $\mathcal{I}' = \mathcal{I}^{\mathcal{E}(\mathsf{eff},\mathcal{I})}$ it follows that $d \in A^{\mathcal{I}'}$.

Now assume $w, \sigma \models A(d) \wedge \neg\big(\gamma_A^-\big)^{x\ a}_{d\ \beta}$. It follows that $w, \sigma \models A(d)$ and $w, \sigma \cdot \beta \models A(d)$ (by assumption). The construction of γ_A^- implies that there is no $\psi/\neg A(d) \in \mathsf{eff}$ with $w, \sigma \models \mathsf{tr}(\psi)$. Again with $\mathcal{I} \cong_{\mathcal{R}} (w, \sigma)$ it follows that $d \in A^{\mathcal{I}}$ and $\neg A(d) \notin \mathcal{E}(\mathsf{eff}, \mathcal{I})$ and therefore $d \in A^{\mathcal{I}'}$.

$\Leftarrow$: To show the other direction we assume $d \in A^{\mathcal{I}'}$ and show $w[A(d), \sigma \cdot \beta] = 1$. Assume $A(d) \in \mathcal{E}(\mathsf{eff}, \mathcal{I})$. It follows that $w, \sigma \models \big(\gamma_A^+\big)^{x\ a}_{d\ \beta}$ and therefore $w[A(d), \sigma \cdot \beta] = 1$ by Lemma 14. Otherwise, if $A(d) \notin \mathcal{E}(\mathsf{eff}, \mathcal{I})$, then $d \in A^{\mathcal{I}'}$ implies $d \in A^{\mathcal{I}}$ and $\neg A(d) \notin \mathcal{E}(\mathsf{eff}, \mathcal{I})$. It follows that $w, \sigma \models A(d) \wedge \neg\big(\gamma_A^-\big)^{x\ a}_{d\ \beta}$ and therefore $w[A(d), \sigma \cdot \beta] = 1$.

For relevant role names $r \in \mathcal{R}$ the proof is analogous. □

In the other direction we construct an interpretation from a given world and an action sequence. For a given world w and $\sigma \in \mathcal{Z}$ an interpretation $\mathcal{I}_{w,\sigma} = (\Delta, \cdot^{\mathcal{I}_{w,\sigma}})$ is defined as follows:

- $A^{\mathcal{I}_{w,\sigma}} := \{d \in \Delta \mid w[A(d), \sigma] = 1\}$ for all $A \in N_C$ and
- $r^{\mathcal{I}_{w,\sigma}} := \{(d, d') \in \Delta \times \Delta \mid w[r(d, d'), \sigma] = 1\}$ for all $r \in N_R$.

Again we omit σ if $\sigma = \langle\rangle$.

Since $\mathcal{D}_{\mathcal{K},\Sigma}$ is objective, there exists a unique epistemic state $e \subseteq W$ such that

$$e \models \mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma})$$

with $e = \{w \mid w \models \mathcal{D}_{\mathcal{K},\Sigma}\}$. In the next lemma we show the relation between e and $\mathcal{M}(\mathcal{K})$.

Lemma 16. *Let $\sigma \in \Sigma_{\mathsf{g}}^* \cap \mathcal{Z}$ and $e = \{w \mid w \models \mathcal{D}_{\mathcal{K},\Sigma}\}$.*

1. *Let $\mathcal{I} \in \mathcal{M}(\mathcal{K})$. It holds that for $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}', \mathcal{W}')$ there exists $w \in e$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w, \sigma)$.*
2. *Let $w \in e$. It holds that here exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}', \mathcal{W}')$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w, \sigma)$.*

Proof. The claim is shown by induction on the length ℓ of the action sequence σ.

$\ell = 0$:

1. We have $\sigma = \langle\rangle$. Let $w \in W$ be arbitrary but fixed and $\mathcal{I} \in \mathcal{M}(\mathcal{K})$. We show that $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \cong_{\mathcal{R}} (e, w_{\mathcal{I}})$: From Lemma 14 it follows that $w_{\mathcal{I}} \in e$ and $\mathcal{I} \cong_{\mathcal{R}} w_{\mathcal{I}}$. By definition it holds that $e^{w_{\mathcal{I}}}_{\langle\rangle} = e$. For all $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ it follows that $w_{\mathcal{J}} \in e$ and $\mathcal{J} \cong_{\mathcal{R}} w_{\mathcal{J}}$. And for all $w' \in e$ it holds by definition that $\mathcal{I}_{w'} \cong_{\mathcal{R}} w'$. Since $w' \models \mathcal{D}_0^{\mathcal{K}}$ we also have $\mathcal{I}_{w'} \in \mathcal{M}(\mathcal{K})$.
2. Let $w \in e$. As argued in the proof of 1. it holds that $\mathcal{I}_w \in \mathcal{M}(\mathcal{K})$. As in the proof of 1. it can be shown that $(\mathcal{I}_w, \mathcal{M}(\mathcal{K})) \cong_{\mathcal{R}} (e, w)$.

$\ell - 1 \to \ell$:

Let $\sigma = \sigma' \cdot \beta$ with ground action $\beta = (\mathsf{eff}, \mathsf{sense} : \{\varsigma\})$.

1. Let $\mathcal{I} \in \mathcal{M}(\mathcal{K})$. By induction there exists $w \in e$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w, \sigma')$ for $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma'} (\mathcal{I}', \mathcal{W}')$. We show that for $(\mathcal{I}'', \mathcal{W}'')$ with $(\mathcal{I}', \mathcal{W}') \Longrightarrow_\beta (\mathcal{I}'', \mathcal{W}'')$ it holds that $(\mathcal{I}'', \mathcal{W}'') \cong_{\mathcal{R}} (e, w, \sigma' \cdot \beta)$:
   - Since $\mathcal{I}'' = \mathcal{I}'^{\mathcal{E}(\mathsf{eff}, \mathcal{I}')}$ and $\mathcal{I}' \cong_{\mathcal{R}} (w, \sigma')$ it follows from Lemma 15 that $\mathcal{I}'' \cong_{\mathcal{R}} (w, \sigma' \cdot \beta)$.
   - Let $\mathcal{J}'' \in \mathcal{W}''$. We need to show that there exists $w'' \in e^w_{\sigma' \cdot \beta}$ such that $\mathcal{J}'' \cong_{\mathcal{R}} w''$. It holds that $\mathcal{J}'' = \mathcal{J}'^{\mathcal{E}(\mathsf{eff}, \mathcal{J}')}$ for an interpretation $\mathcal{J}' \in \mathcal{W}'$. By induction there exists $w' \in e^w_{\sigma}$ such that $\mathcal{J}' \cong_{\mathcal{R}} w'$. By Lemma 15 it follows that $\mathcal{J}'' \cong_{\mathcal{R}} (w', \beta)$. We show that $w'_\beta \in e^w_{\sigma' \cdot \beta}$. For this, it is sufficient to show $w' \simeq_\beta w_{\sigma'}$, i.e. $w'[SF(\beta), \langle\rangle] = w_{\sigma'}[SF(\beta), \langle\rangle]$. To show this we proceed as follows:
     Since $\mathcal{J}'^{\mathcal{E}(\mathsf{eff}, \mathcal{J}')} \in \mathcal{W}''$ it follows that $\mathcal{I}' \sim_\beta \mathcal{J}'$. With $\mathsf{sense} = \{\varsigma\}$ it follows that $\mathcal{I}' \models \varsigma$ iff $\mathcal{J}' \models \varsigma$. We have $\mathcal{J}' \cong_{\mathcal{R}} w'$ and $\mathcal{I}' \cong_{\mathcal{R}} (w, \sigma')$. Therefore, since ς is objective and with Lemma 13 we get $w' \models \mathsf{tr}(\varsigma)$ iff $\mathcal{J}' \models \varsigma$ and $w, \sigma' \models \mathsf{tr}(\varsigma)$ iff $\mathcal{I}' \models \varsigma$. Consequently, $w' \models \mathsf{tr}(\varsigma)$ iff $w_{\sigma'} \models \mathsf{tr}(\varsigma)$. By assumption both w' and $w_{\sigma'}$ are the progression of a world that satisfies the BAT $\mathcal{D}_{\mathcal{K},\Sigma}$. With $w' \models \mathsf{tr}(\varsigma)$ iff $w_{\sigma'} \models \mathsf{tr}(\varsigma)$ and by construction of $\mathcal{D}^{\Sigma}_{\text{sense}}$ it follows that $w' \models \left(\gamma_{SF}\right)^a_\beta$ iff $w_{\sigma'} \models \left(\gamma_{SF}\right)^a_\beta$ and therefore $w'[SF(\beta), \langle\rangle] = w_{\sigma'}[SF(\beta), \langle\rangle]$. This implies $w'_\beta \in e^w_{\sigma' \cdot \beta}$ and we get $\mathcal{J}'' \cong_{\mathcal{R}} w''$ with $w'' = w'_\beta$.

- Let $w'' \in e^{w}_{\sigma' \cdot \beta}$. We need to show that there exists $\mathcal{J}'' \in \mathcal{W}''$ such that $\mathcal{J}'' \cong_{\mathcal{R}} w''$. By definition of the progression of an epistemic state we have

  $$w'' \in \{\hat{w}_{\sigma' \cdot \beta} \mid \hat{w} \in e, \hat{w} \simeq_{\sigma' \cdot \beta} w\}.$$

  Thus, there exists a world $w' \in e^{w}_{\sigma'}$ such that $w'' = w'_{\beta}$. Since by induction we have $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w, \sigma')$, there exists $\mathcal{J}' \in \mathcal{W}'$ such that $\mathcal{J}' \cong_{\mathcal{R}} w'$. By applying Lemma 15 we get $\mathcal{J}'^{\mathcal{E}(\mathsf{eff}, \mathcal{J}')} \cong_{\mathcal{R}} (w', \beta)$ and therefore $\mathcal{J}'^{\mathcal{E}(\mathsf{eff}, \mathcal{J}')} \cong_{\mathcal{R}} w'_{\beta}$. We need to show that $\mathcal{J}'^{\mathcal{E}(\mathsf{eff}, \mathcal{J}')} \in \mathcal{W}''$. This holds iff $\mathcal{I}' \sim_{\beta} \mathcal{J}'$. We have that $w' \simeq_{\beta} w_{\sigma'}$. And since $\mathcal{I}' \cong_{\mathcal{R}} w_{\sigma'}$ and $\mathcal{J}' \cong_{\mathcal{R}} w'$, it can be shown as in the previous item that $\mathcal{I}' \sim_{\beta} \mathcal{J}'$.

2. Let $w \in e$. By induction there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma'} (\mathcal{I}', \mathcal{W}')$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w, \sigma')$. As in the proof of 1. it can be shown that for $(\mathcal{I}'', \mathcal{W}'')$ with $(\mathcal{I}', \mathcal{W}') \Longrightarrow_{\beta} (\mathcal{I}'', \mathcal{W}'')$ it holds that $(\mathcal{I}'', \mathcal{W}'') \cong_{\mathcal{R}} (e, w, \sigma' \cdot \beta)$.

□

Now we are ready to show the correctness of the construction as a consequence of Lemma 16.

Theorem 17. *Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, Σ a set of primitive actions, $\sigma \in \Sigma_{\mathsf{g}}^*$ a sequence of ground actions, ψ an $\mathcal{ALCOK}$-ground formula or axiom and $\mathcal{D}_{\mathcal{K},\Sigma}$ the BAT constructed as described above. It holds that ψ is valid after executing σ in $\mathcal{K}$ iff $\mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma} \models [\sigma]\mathsf{tr}(\psi)$.*

Proof.

$\Rightarrow$: Assume to the contrary that ψ is valid after executing σ in $\mathcal{K}$ and there exists a world w' such that $e, w' \models \mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma}$ and $e, w', \sigma \not\models \mathsf{tr}(\psi)$. It holds that $e = \{w \mid w \models \mathcal{D}_{\mathcal{K},\Sigma}\}$ and $w' \in e$. By Lemma 16 there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma} (\mathcal{I}', \mathcal{W}')$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w', \sigma)$. Since by assumption $e, w', \sigma \not\models \mathsf{tr}(\psi)$ it follows from Lemma 13 that $(\mathcal{I}', \mathcal{W}') \not\models\!\!\!\models \psi$ which is a contradiction to the assumption that ψ is valid in $\mathcal{K}$ after executing σ.

$\Leftarrow$: Assume to the contrary that $\mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma} \models [\sigma]\mathsf{tr}(\psi)$ and there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $(\mathcal{I}', \mathcal{W}')$ with $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma} (\mathcal{I}', \mathcal{W}')$ such that $(\mathcal{I}', \mathcal{W}') \not\models\!\!\!\models \psi$. By Lemma 16 there exists $w' \in e = \{w \mid w \models \mathcal{D}_{\mathcal{K},\Sigma}\}$ such that $(\mathcal{I}', \mathcal{W}') \cong_{\mathcal{R}} (e, w', \sigma)$. By Lemma 13 we therefore have $e, w', \sigma \not\models \mathsf{tr}(\psi)$ which is a contradiction to the assumption that $\mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma} \models [\sigma]\mathsf{tr}(\psi)$ because $e, w' \models \mathit{OKnow}(\mathcal{D}_{\mathcal{K},\Sigma}) \wedge \mathcal{D}_{\mathcal{K},\Sigma}$.

□

3.3 Deciding the Projection Problem

In Theorem 17 we have characterized the projection problem as a standard entailment problem in the epistemic situation calculus. The Representation Theorem for $\mathcal{ES}$ [LL04] provides us with a method for reducing projection to standard (non-modal) first-order reasoning by eliminating the action and knowledge modalities in the projection query. For the action modality regression is used to obtain a sentence that refers only to the initial situation. Given the initial KB subformulas of the form $\mathit{Know}(\phi)$ are then replaced by objective formulas ϕ' that capture the known instances of ϕ w.r.t. the initial KB. To obtain a decision procedure for the projection problem we show that a similar reduction can be done within $\mathcal{ALCO}$. We combine the reduction approach used in [BLM$^+$05] for the non-epistemic projection problem and a method for rewriting subjective concepts to objective ones in the projection query resembling the Representation Theorem [Lev84, LL01] in presence of only-knowing. We show that the projection problem is ExpTime-complete.

First, we consider some basic properties of knowledge states that evolve from the initial epistemic model by executing a sequence of primitive ground actions. In the following Ind denotes the finite set of all individuals that are mentioned in the input, i.e. we assume that all concepts, axioms, formulas, KBs and primitive actions use only individuals from Ind. The elements in Ind are called *named elements* and the ones in $\Delta \setminus \mathsf{Ind}$ are called *anonymous* or *unnamed elements*.

Lemma 18. *Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ an interpretation, σ a sequence of primitive ground actions, D an $\mathcal{ALCO}$-concept and $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation such that $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$. It holds that $\big(\boldsymbol{K}D\big)^{\mathcal{W}_n} \cap \big(\Delta \setminus \mathsf{Ind}\big) \neq \emptyset$ implies $\Delta \setminus \mathsf{Ind} \subseteq \big(\boldsymbol{K}D\big)^{\mathcal{W}_n}$.*

We first present a short outline of the proof. Donini et. al [DLN+98] showed that the interpretations contained in the epistemic model $\mathcal{M}(\mathcal{K})$ of an $\mathcal{ALC}$-KB $\mathcal{K}$ are closed under renaming of anonymous elements. As the term "anonymous" suggests anonymous elements are indistinguishable. We use this observation for the proof of Lemma 18 as follows. Assume to the contrary that there exist two unnamed elements $d, e \in \Delta \setminus \mathsf{Ind}$ such that $d \in \big(\mathbf{K}D\big)^{\mathcal{W}_n}$ and $e \notin \big(\mathbf{K}D\big)^{\mathcal{W}_n}$. We show that there exists an interpretation $\mathcal{J}_n \in \mathcal{W}_n$ such that $d \in D^{\mathcal{J}_n}$ but $e \notin D^{\mathcal{J}_n}$. To obtain the contradiction we construct an interpretation $\mathcal{J}'_n$ also contained in $\mathcal{W}_n$ with $d \notin D^{\mathcal{J}'_n}$ by just "swapping" the two unnamed elements d and e in $\mathcal{J}_n$.

First, we define the renaming of an interpretation.

Definition 19 (renamed interpretation)**.** Let $\mathcal{Y} = (\Delta^{\mathcal{Y}}, \cdot^{\mathcal{Y}})$ be an interpretation with a countably infinite domain $\Delta^{\mathcal{Y}}$ and a mapping $\cdot^{\mathcal{Y}}$ that satisfies $a^{\mathcal{Y}} \neq b^{\mathcal{Y}}$ for all $a, b \in N_I$. As before $\Delta := N_I$ denotes the domain of standard names. Let $\iota : \Delta^{\mathcal{Y}} \to \Delta$ be a bijection. The *renamed interpretation of $\mathcal{Y}$ with ι*, denoted by $\iota(\mathcal{Y})$, is defined as follows:

$$\begin{aligned} a^{\iota(\mathcal{Y})} &:= a \text{ for all } a \in N_I; \\ A^{\iota(\mathcal{Y})} &:= \{\iota(d) \mid d \in A^{\mathcal{Y}}\} \text{ for all } A \in N_C; \\ r^{\iota(\mathcal{Y})} &:= \{(\iota(d), \iota(e)) \mid (d, e) \in r^{\mathcal{Y}}\} \text{ for all } r \in N_R. \end{aligned}$$

▲

We show that updating an interpretation and its renamed version again yields isomorphic interpretations if names of named elements are fixed.

Proposition 20. *Let $\mathcal{I}$ be an interpretation, $\iota : \Delta \to \Delta$ a bijection with $\iota(a) = a$ for all $a \in \mathsf{Ind}$, L a set of literals, C an $\mathcal{ALCO}$-concept, ϱ an $\mathcal{ALCO}$-axiom and $\mathcal{J} := \iota(\mathcal{I})$.*

1. *$d \in A^{\mathcal{I}^{\mathsf{L}}}$ iff $\iota(d) \in A^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$ and $A \in N_C$;*
2. *$(d, e) \in r^{\mathcal{I}^{\mathsf{L}}}$ iff $(\iota(d), \iota(e)) \in r^{\mathcal{J}^{\mathsf{L}}}$ for all $d, e \in \Delta$ and $r \in N_R$;*
3. *$d \in C^{\mathcal{I}^{\mathsf{L}}}$ iff $\iota(d) \in C^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$;*
4. *$\mathcal{I}^{\mathsf{L}} \models \varrho$ iff $\mathcal{J}^{\mathsf{L}} \models \varrho$.*

Proof.

1. Let $A \in N_C$ and $d \in \Delta$. We show $d \in A^{\mathcal{I}^{\mathsf{L}}}$ iff $\iota(d) \in A^{\mathcal{J}^{\mathsf{L}}}$. By the definition of renamed interpretations we have $d \in A^{\mathcal{I}}$ iff $\iota(d) \in A^{\mathcal{J}}$.

   Furthermore it holds that $d \in \{a \mid \neg A(a) \in \mathsf{L}\}$

   iff there exists $b \in \mathsf{Ind}$ such that $d = b$ and $\neg A(b) \in \mathsf{L}$, since by assumption it holds that $\{a \mid \neg A(a) \in \mathsf{L}\} \subseteq \mathsf{Ind}$

   iff $\iota(d) = b$ and $\neg A(b) \in \mathsf{L}$, by assumption on ι

   iff $\iota(d) \in \{a \mid \neg A(a) \in \mathsf{L}\}$.

   In the same way it can be shown that $d \in \{a \mid A(a) \in \mathsf{L}\}$ iff $\iota(d) \in \{a \mid A(a) \in \mathsf{L}\}$. By the definition of interpretation updates it now follows that $d \in A^{\mathcal{I}^{\mathsf{L}}}$ iff $\iota(d) \in A^{\mathcal{J}^{\mathsf{L}}}$.

2. The proof is analogous to the proof of claim 1.

3. The claim is proven by induction on the structure of C.

   $C = A :$ for some $A \in N_C$. See proof of claim 1.

   $C = \{a\} :$ for some $a \in \mathsf{Ind}$. We get $d \in \{a\}^{\mathcal{I}^{\mathsf{L}}}$ iff $d \in \{a^{\mathcal{I}^{\mathsf{L}}}\}$ iff $d = a$ iff $\iota(d) = a$ iff $\iota(d) \in \{a^{\mathcal{J}^{\mathsf{L}}}\}$ iff $\iota(d) \in \{a\}^{\mathcal{J}^{\mathsf{L}}}$.

   We omit the proof of the remaining cases.

4. The claim directly follows from the other three claims.

□

For the proof of Lemma 18 we need another auxiliary proposition that is a direct consequence of the one shown above.

Proposition 21. *Let* $(\mathcal{I}_0, \mathcal{W}_0) \Longrightarrow_{\alpha_0} (\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\alpha_1} \cdots \Longrightarrow_{\alpha_{n-1}} (\mathcal{I}_n, \mathcal{W}_n)$ *be a sequence of epistemic interpretations with* $\mathcal{W}_0 = \mathcal{M}(\mathcal{K})$ *for an* $\mathcal{ALC}$*-KB* $\mathcal{K}$ *and* $\alpha_j = (\mathsf{eff}_j, \mathsf{sense}_j)$ *with* $j = 0, \ldots, n-1$. *Let* $\mathcal{Y}_0, \mathcal{Y}_1, \ldots, \mathcal{Y}_n$ *be a sequence of interpretations such that*

- $\mathcal{Y}_i \in \mathcal{W}_i$ *for all* $i = 0, \ldots, n$ *and*
- $\mathcal{Y}_{j+1} = \mathcal{Y}_j^{\mathcal{E}(\mathsf{eff}_j, \mathcal{Y}_j)}$ *for all* $j = 0, \ldots, n-1$.

Furthermore, let $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ *be an interpretations such that for all sets of literals* L *and all* $\mathcal{ALCO}$*-ground formulas* φ *it holds that* $\mathcal{Y}_0{}^{\mathsf{L}} \models \psi$ *iff* $\mathcal{J}_0{}^{\mathsf{L}} \models \psi$. *The claim is that for each* $i \in \{0, \ldots, n\}$ *there exists a set of literals* L_i *such that* $\mathcal{J}_0{}^{\mathsf{L}_i} \in \mathcal{W}_i$ *and* $\mathcal{Y}_i = \mathcal{Y}_0{}^{\mathsf{L}_i}$.

Proof. We prove it by induction on n.

$n = 1 :$ For $i = 0$ the claim trivially holds for $\mathsf{L}_0 = \emptyset$. Let $i := 1$. By definition it holds that $\mathcal{Y}_1 = \mathcal{Y}_0{}^{\mathcal{E}(\mathsf{eff}_0, \mathcal{Y}_0)}$. By assumption we have $\mathcal{Y}_0 \models \psi$ iff $\mathcal{J}_0 \models \psi$ for all ground formulas occurring in α_0. Hence, $\mathcal{Y}_0 \sim_{\alpha_0} \mathcal{J}_0$ and $\mathcal{E}(\mathsf{eff}_0, \mathcal{Y}_0) = \mathcal{E}(\mathsf{eff}_0, \mathcal{J}_0)$. Consequently, $\mathcal{J}_0{}^{\mathcal{E}(\mathsf{eff}_0, \mathcal{J}_0)} \in \mathcal{W}_1$.

$n - 1 \to n :$ We assume that there exists a set of literals L such that $\mathcal{Y}_{n-1} = \mathcal{Y}_0{}^{\mathsf{L}}$ and $\mathcal{J}_0{}^{\mathsf{L}} \in \mathcal{W}_{n-1}$. Due to the assumption we have $\mathcal{Y}_{n-1} \models \psi$ iff $\mathcal{J}_{n-1} \models \psi$ for any $\mathcal{ALCO}$-ground formula ψ. Consequently, $\mathcal{E}(\mathsf{eff}_{n-1}, \mathcal{Y}_{n-1}) = \mathcal{E}(\mathsf{eff}_{n-1}, \mathcal{J}_0{}^{\mathsf{L}})$ and $\mathcal{Y}_0 \sim_{\alpha_{n-1}} \mathcal{J}_0{}^{\mathsf{L}}$. We obtain $\mathcal{Y}_n = \mathcal{Y}_0{}^{\mathsf{L}'}$ and $\mathcal{J}_0{}^{\mathsf{L}'} \in \mathcal{W}_n$ with $\mathsf{L}' = \mathsf{L} \setminus \neg\mathcal{E}(\mathsf{eff}_{n-1}, \mathcal{Y}_{n-1}) \cup \mathcal{E}(\mathsf{eff}_{n-1}, \mathcal{Y}_{n-1})$ due to Lemma 2.

□

Now we are ready to prove Lemma 18.

Proof of Lemma 18. Let $(\mathcal{I}_0, \mathcal{W}_0) \Longrightarrow_{\alpha_0} (\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\alpha_1} \cdots \Longrightarrow_{\alpha_{n-1}} (\mathcal{I}_n, \mathcal{W}_n)$ be the sequence of epistemic interpretations obtained by executing $\sigma = \alpha_0, \ldots, \alpha_{n-1}$ in $(\mathcal{I}_0, \mathcal{M}(\mathcal{K}))$. Assume $\big(\mathsf{K}D\big)^{\mathcal{W}_n} \cap \big(\Delta \backslash \mathsf{Ind}\big) \neq \emptyset$ for an $\mathcal{ALCO}$-concept D. We need to show that $\Delta \backslash \mathsf{Ind} \subseteq \big(\mathsf{K}D\big)^{\mathcal{W}_n}$. Assume to the contrary that there is an anonymous elements $e \in \Delta \setminus \mathsf{Ind}$ such that $e \notin \mathsf{K}D^{\mathcal{W}_n}$. Consequently, there exists a sequence of interpretations $\mathcal{Y}_0, \mathcal{Y}_1, \ldots, \mathcal{Y}_n$ such that $\mathcal{Y}_i \in \mathcal{W}_i$ for all $i = 0, \ldots, n$, $\mathcal{Y}_{j+1} = \mathcal{Y}_j^{\mathcal{E}(\mathsf{eff}_j, \mathcal{Y}_j)}$ for all $j = 0, \ldots, n-1$ and $e \notin D^{\mathcal{Y}_n}$.

Next we choose an anonymous element $d \in \Delta \setminus \mathsf{Ind}$ with $d \in \mathsf{K}D^{\mathcal{W}_n}$. By assumption such an element exists. Furthermore, we choose a bijection $\iota : \Delta \to \Delta$ such that $\iota(a) = a$ for all $a \in \mathsf{Ind}$ and $\iota(e) = d$ and $\iota(d) = e$. Let $\mathcal{J}_0 := \iota(\mathcal{Y}_0)$. Using Proposition 20.4 it follows that for all set of literals L and all $\mathcal{ALCO}$-axioms ϱ it holds that ${\mathcal{Y}_0}^{\mathsf{L}} \models \varrho$ iff ${\mathcal{J}_0}^{\mathsf{L}} \models \varrho$ and $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ as required for Proposition 21. Therefore, Proposition 21 implies that there exists a set of literals L such that ${\mathcal{J}_0}^{\mathsf{L}} \in \mathcal{W}_n$ and $\mathcal{Y}_n = {\mathcal{Y}_0}^{\mathsf{L}}$. Let $\mathcal{J}_n := {\mathcal{Y}_0}^{\mathsf{L}}$. With Proposition 20.3, $e \notin D^{\mathcal{Y}_n}$ implies $\iota(e) \notin D^{\mathcal{J}_n}$. Since $\iota(e) = d$ and $\mathcal{J}_n \in \mathcal{W}_n$ it follows that $d \notin \mathsf{K}D^{\mathcal{W}_n}$ which is a contradiction. □

For epistemic concepts where the concept under the scope of the K-constructor is K-free and in addition also nominal-free we can even show a simpler property as well as for epistemic roles.

Lemma 22. *Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ an interpretation, $\sigma = \alpha_0, \ldots, \alpha_{n-1}$ a sequence of primitive ground actions, C an $\mathcal{ALC}$-concept, $r \in N_R$ and $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation with $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$. It holds that*

1. *if $\mathcal{K} \not\models \top \sqsubseteq C$, then $(\mathsf{K}C)^{\mathcal{W}_n} \subseteq \mathsf{Ind}$ and*
2. *$(\mathsf{K}r)^{\mathcal{W}_n} \subseteq \mathsf{Ind} \times \mathsf{Ind}$.*

Before we prove this lemma we consider the static case where the action sequence is empty. A proof for this case was given by Mehdi [Meh14] (page 76, Lemma 4) also using the swapping technique for unnamed elements from [DLN$^+$98]. Mehdi's proof can be outlined as follows. Assume to the contrary that there exists an unnamed element $d \in \Delta \setminus \mathsf{Ind}$ such that $d \in (\mathsf{K}C)^{\mathcal{M}(\mathcal{K})}$. Since $\mathcal{K} \not\models \top \sqsubseteq C$, there exists a model $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ and an anonymous element $e \in \Delta \setminus \mathsf{Ind}$ such that $e \notin C^{\mathcal{J}}$. Now consider a bijection $\iota : \Delta \to \Delta$ with $\iota(e) = d$ and $\iota(d) = e$ and $\iota(a) = a$ for all $a \in \mathsf{Ind}$. The renaming ι just swaps the two unnamed elements d and e. The renaming of the model $\mathcal{J}$ using ι yields an interpretation, denoted by $\iota(\mathcal{J})$, that is isomorphic to $\mathcal{J}$ and is also a model of $\mathcal{K}$. Thus, from $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ and $e \notin C^{\mathcal{J}}$ it follows that $\iota(\mathcal{J}) \in \mathcal{M}(\mathcal{K})$ and $\iota(e) \notin C^{\iota(\mathcal{J})}$. With $\iota(e) = d$ we get a contradiction to the assumption $d \in (\mathsf{K}C)^{\mathcal{M}(\mathcal{K})}$. To reuse this idea for the dynamic case we need to show that if initially $\mathcal{K} \not\models \top \sqsubseteq C$ holds, then we can still find in $\mathcal{W}_n$ an interpretation $\mathcal{J}$ such that there exists an unnamed domain element *not* contained in the extension of C under $\mathcal{J}$. This domain element will serve as the "swap partner" for the unnamed known instance d of $\mathsf{K}D$ under $\mathcal{W}_n$ and will lead to the contradiction as in the static case.

For the construction of such an interpretation $\mathcal{J} \in \mathcal{W}_n$ we introduce an operation that merges two interpretations together in one.

Definition 23. Let $\mathcal{I}_0$ and $\mathcal{I}_1$ be two interpretations. The *sum of $\mathcal{I}_0$ and $\mathcal{I}_1$* is an interpretation, denoted by $\mathcal{I}_0 \oplus \mathcal{I}_1$, that is defined as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}_0 \oplus \mathcal{I}_1} &:= \Delta \times \{0,1\}; \\
A^{\mathcal{I}_0 \oplus \mathcal{I}_1} &:= \{\langle d,0\rangle \mid d \in A^{\mathcal{I}_0}\} \cup \{\langle d,1\rangle \mid d \in A^{\mathcal{I}_1}\} \text{ for all } A \in N_C \\
r^{\mathcal{I}_0 \oplus \mathcal{I}_1} &:= \{(\langle d,0\rangle, \langle e,0\rangle) \mid (d,e) \in r^{\mathcal{I}_0}\} \cup \{(\langle d,1\rangle, \langle e,1\rangle) \mid (d,e) \in r^{\mathcal{I}_1}\} \text{ for all } r \in N_R \\
a^{\mathcal{I}_0 \oplus \mathcal{I}_1} &:= \langle a,0\rangle \text{ for all } a \in N_I.
\end{aligned}$$

▲

Note that the operation is non-commutative due to the interpretation of individual constants. We now consider a renaming of the sum $\mathcal{I}_0 \oplus \mathcal{I}_1$ that interprets the named part of the domain given by Ind as in $\mathcal{I}_0$.

Proposition 24. *Let $\mathcal{I}_0$, $\mathcal{I}_1$ and $\mathcal{I}_0 \oplus \mathcal{I}_1$ be as above, $\iota : \Delta \times \{0,1\} \to \Delta$ a bijection such that $\iota(\langle a,0\rangle) = a$ for all $a \in \mathsf{Ind}$, L a set of literals, C an $\mathcal{ALCO}$-concept, φ an $\mathcal{ALCO}$-ABox assertion and $\mathcal{J} := \iota(\mathcal{I}_0 \oplus \mathcal{I}_1)$.*

1. *$d \in A^{\mathcal{I}_0{}^{\mathsf{L}}}$ iff $\iota(\langle d,0\rangle) \in A^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$ and $A \in N_C$;*
2. *$(d,e) \in r^{\mathcal{I}_0{}^{\mathsf{L}}}$ iff $\big(\iota(\langle d,0\rangle), \iota(\langle e,0\rangle)\big) \in r^{\mathcal{J}^{\mathsf{L}}}$ for all $d,e \in \Delta$ and $r \in N_R$;*
3. *$d \in C^{\mathcal{I}_0{}^{\mathsf{L}}}$ iff $\iota(\langle d,0\rangle) \in C^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$;*
4. *${\mathcal{I}_0}^{\mathsf{L}} \models \varphi$ iff $\mathcal{J}^{\mathsf{L}} \models \varphi$.*

Proof.

1. Let $d \in \Delta$ and $A \in N_C$ and L a set of literals. First we show
$$d \in A^{\mathcal{I}_0} \text{ iff } \iota(\langle d,0\rangle) \in A^{\mathcal{J}}. \tag{1}$$
Using the definitions we get $d \in A^{\mathcal{I}_0}$ iff $\langle d,0\rangle \in A^{\mathcal{I}_0 \oplus \mathcal{I}_1}$ iff $\iota(\langle d,0\rangle) \in A^{\iota(\mathcal{I}_0 \oplus \mathcal{I}_1)}$. Since L contains only individuals from Ind and by construction $\iota(\langle a,0\rangle) = a$ for all $a \in \mathsf{Ind}$, it follows that
$$d \in \{a \mid A(a) \in \mathsf{L}\} \text{ iff } \iota(\langle d,0\rangle) \in \{a \mid A(a) \in \mathsf{L}\} \tag{2}$$
and
$$d \in \{a \mid \neg A(a) \in \mathsf{L}\} \text{ iff } \iota(\langle d,0\rangle) \in \{a \mid \neg A(a) \in \mathsf{L}\}. \tag{3}$$
By definition of interpretation update and (1), (2) and (3) it follows that $d \in A^{\mathcal{I}_0{}^{\mathsf{L}}}$ iff $\iota(\langle d,0\rangle) \in A^{\mathcal{J}^{\mathsf{L}}}$.
2. The proof is analogous to the proof of 1.
3. The proof is by induction on the structure of C.

   $C = A :$ for some $A \in N_C$, see 1.

   $C = \{b\}$: for some $b \in \mathsf{Ind}$. It holds that $d \in \{b\}^{\mathcal{I}_0{}^{\mathsf{L}}}$
   iff $d \in \{b^{\mathcal{I}_0{}^{\mathsf{L}}}\}$
   iff $d \in \{b\}$
   iff $d = b$

iff $\iota(\langle d,0\rangle) = b$
iff $\iota(\langle d,0\rangle) \in \{b\}$
iff $\iota(\langle d,0\rangle) \in \{b^{\mathcal{J}^{\mathsf{L}}}\}$
iff $\iota(\langle d,0\rangle) \in \{b\}^{\mathcal{J}^{\mathsf{L}}}$.

$C = \neg D$: It holds that $d \in (\neg D)^{\mathcal{I}_0{}^{\mathsf{L}}}$

iff $d \notin D^{\mathcal{I}_0{}^{\mathsf{L}}}$
iff $\iota(\langle d,0\rangle) \notin D^{\mathcal{J}^{\mathsf{L}}}$ (by induction)
iff $\iota(\langle d,0\rangle) \in (\neg D)^{\mathcal{J}^{\mathsf{L}}}$.

$C = D_1 \sqcap D_2$: It holds that $d \in (D_1 \sqcap D_2)^{\mathcal{I}_0{}^{\mathsf{L}}}$

iff $d \in D_1^{\mathcal{I}_0{}^{\mathsf{L}}}$ and $d \in D_2^{\mathcal{I}_0{}^{\mathsf{L}}}$
iff $\iota(\langle d,0\rangle) \in D_1^{\mathcal{J}^{\mathsf{L}}}$ and $\iota(\langle d,0\rangle) \in D_2^{\mathcal{J}^{\mathsf{L}}}$ (by induction)
iff $\iota(\langle d,0\rangle) \in (D_1 \sqcap D_2)^{\mathcal{J}^{\mathsf{L}}}$.

$C = \exists r.D$: It holds that $d \in (\exists r.D)^{\mathcal{I}_0{}^{\mathsf{L}}}$

iff there exists an $e \in \Delta$ s.t. $(d,e) \in r^{\mathcal{I}_0{}^{\mathsf{L}}}$ and $e \in D^{\mathcal{I}_0{}^{\mathsf{L}}}$
iff $\big(\iota(\langle d,0\rangle), \iota(\langle e,0\rangle)\big) \in r^{\mathcal{J}^{\mathsf{L}}}$ (by claim 2.) and $\iota(\langle e,0\rangle) \in D^{\mathcal{J}_0{}^{\mathsf{L}}}$ (by induction)
iff $\iota(\langle d,0\rangle) \in (\exists r.D)^{\mathcal{J}^{\mathsf{L}}}$.

The last equivalence holds because $(d,e) \in r^{\mathcal{J}^{\mathsf{L}}}$ and $\iota^-(d) = \langle d',0\rangle$ for some $d' \in \Delta$ implies $\iota^-(e) = \langle e',0\rangle$ for some $e' \in \Delta$.

4. It follows from claim 2 and 3 and the fact that $\iota(\langle a,0\rangle) = a$ for all $a \in \mathsf{Ind}$.

□

Intuitively, the $\mathcal{I}_0$-part of $\iota(\mathcal{I}_0 \oplus \mathcal{I}_1)$ behaves like $\mathcal{I}_0$ whereas the $\mathcal{I}_1$-part remains unchanged as shown in the next proposition.

Proposition 25. *Let $\mathcal{I}_0$, $\mathcal{I}_1$ and $\mathcal{I}_0 \oplus \mathcal{I}_1$ be as above, $\iota : \Delta \times \{0,1\} \to \Delta$ be a bijection such that $\iota(\langle a,0\rangle) = a$ for all $a \in \mathsf{Ind}$, L a set of literals, C an $\mathcal{ALC}$-concept and $\mathcal{J} := \iota(\mathcal{I}_0 \oplus \mathcal{I}_1)$.*

1. *$d \in A^{\mathcal{I}_1}$ iff $\iota(\langle d,1\rangle) \in A^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$ and $A \in N_C$;*
2. *$(d,e) \in r^{\mathcal{I}_1}$ iff $\big(\iota(\langle d,1\rangle), \iota(\langle e,1\rangle)\big) \in r^{\mathcal{J}^{\mathsf{L}}}$ for all $d,e \in \Delta$ and $r \in N_R$;*
3. *$d \in C^{\mathcal{I}_1}$ iff $\iota(\langle d,1\rangle) \in C^{\mathcal{J}^{\mathsf{L}}}$ for all $d \in \Delta$.*

Proof.

1. Due to the construction of $\mathcal{J} := \iota(\mathcal{I}_0 \oplus \mathcal{I}_1)$ it holds that $d \in A^{\mathcal{I}_1}$ iff $\langle d,1\rangle \in A^{\mathcal{I}_0 \oplus \mathcal{I}_1}$ iff $\iota(\langle d,1\rangle) \in A^{\iota(\mathcal{I}_0 \oplus \mathcal{I}_1)}$. By definition of ι it holds that $\iota(\langle d,1\rangle) \notin \mathsf{Ind}$ for all $d \in \Delta$. Since L contains only individuals from Ind, it follows that $\iota(\langle d,1\rangle) \in A^{\mathcal{J}}$ iff $\iota(\langle d,1\rangle) \in A^{\mathcal{J}^{\mathsf{L}}}$. Consequently, $d \in A^{\mathcal{I}_1}$ iff $\iota(\langle d,1\rangle) \in A^{\mathcal{J}^{\mathsf{L}}}$.
2. The proof is analogous to the proof of claim 1.
3. The proof is by induction on the structure of the $\mathcal{ALC}$-concept C using claim 1 and 2 and the property that for all $(d,e) \in \Delta \times \Delta$ it holds that $(d,e) \in r^{\mathcal{J}^{\mathsf{L}}}$ and $\iota^-(d) = \langle d',1\rangle$ for some $d' \in \Delta$ implies $\iota^-(e) = \langle e',1\rangle$ for some $e' \in \Delta$.

□

Now we are ready to proof Lemma 22.

Proof of Lemma 22. Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$, σ an action sequence, $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation with $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$, C an $\mathcal{ALC}$-concept with $\mathcal{K} \not\models \top \sqsubseteq C$ and $r \in N_R$.

1. We have to show that $\big(\mathsf{K}C\big)^{\mathcal{W}_n} \subseteq \mathsf{Ind}$. Let $d \in \big(\mathsf{K}C\big)^{\mathcal{W}_n}$. Assume to the contrary that $d \in \Delta \setminus \mathsf{Ind}$. Since $\mathcal{K} \not\models \top \sqsubseteq C$, there exists $\mathcal{Y}_0 \in \mathcal{M}(\mathcal{K})$ and $e \in \Delta$ such that $e \notin C^{\mathcal{Y}_0}$. Consider the sum $\mathcal{I}_0 \oplus \mathcal{Y}_0$. Obviously, there exists a bijection $\iota : \Delta \times \{0,1\} \to \Delta$ such that $\iota(\langle a, 0\rangle) = a$ for all $a \in \mathsf{Ind}$ and $\iota(\langle e, 1\rangle) = d$. Let $\mathcal{J} := \iota(\mathcal{I}_0 \oplus \mathcal{Y}_0)$ be the corresponding renamed interpretation. From Proposition 24 and 25 it follows that $\mathcal{J} \in \mathcal{M}(\mathcal{K})$. Proposition 24 implies that ${\mathcal{I}_0}^{\mathsf{L}} \models \psi$ iff $\mathcal{J}^{\mathsf{L}} \models \psi$ for all sets of literals L and all $\mathcal{ALCO}$-ground formulas ψ. Due to Proposition 21 there exists a set of literals L such that $\mathcal{J}^{\mathsf{L}} \in \mathcal{W}_n$ and $\mathcal{I}_n = {\mathcal{I}_0}^{\mathsf{L}}$. Using Proposition 25.3 it holds that $e \in (\neg C)^{\mathcal{Y}_0}$ implies $\iota(\langle e, 1\rangle) \in (\neg C)^{\mathcal{J}^{\mathsf{L}}}$. Since $\iota(\langle e, 1\rangle) = d$ and $\mathcal{J}^{\mathsf{L}} \in \mathcal{W}_n$, we have a contradiction to the assumption $d \in \big(\mathsf{K}C\big)^{\mathcal{W}_n}$ and $d \notin \mathsf{Ind}$.
2. Let $(d, e) \in (\mathsf{K}r)^{\mathcal{W}_n}$. We show that $(d, e) \in \mathsf{Ind} \times \mathsf{Ind}$. First, assume to the contrary that $d \in \Delta \setminus \mathsf{Ind}$. Let $\mathcal{Y}_0 \in \mathcal{M}(\mathcal{K})$ be an arbitrary model and $\mathcal{I}_0 \oplus \mathcal{Y}_0$ the sum of $\mathcal{I}_0$ and $\mathcal{Y}_0$. Since $d \notin \mathsf{Ind}$, there exists a bijection $\iota : \Delta \times \{0,1\} \to \Delta$ such that $\iota(\langle a, 0\rangle) = a$ for all $a \in \mathsf{Ind}$, $\iota(\langle d, 1\rangle) = d$ and $\iota(\langle e, 0\rangle) = e$. Let $\mathcal{J} := \mathcal{I}_0 \oplus \mathcal{Y}_0$ be the corresponding renamed interpretation. As in the first part of the lemma we can show that there exists a set of literals L such that $\mathcal{J}^{\mathsf{L}} \in \mathcal{W}_n$. By definition of the sum it holds that $\big(\langle d, 1\rangle, \langle e, 0\rangle\big) \notin r^{\mathcal{I}_0 \oplus \mathcal{Y}_0}$ which implies also $\big(\iota(\langle d, 1\rangle), \iota(\langle e, 0\rangle)\big) \notin r^{\iota(\mathcal{I}_0 \oplus \mathcal{Y}_0)}$. The bijection ι is defined such that $\iota(\langle d, 1\rangle) \notin \mathsf{Ind}$. Therefore,
$$\big(\iota(\langle d, 1\rangle), \iota(\langle e, 0\rangle)\big) \notin \{(a, b) \mid r(a, b) \in \mathsf{L}\} \subseteq \mathsf{Ind} \times \mathsf{Ind}.$$
Consequently, we have $\big(\iota(\langle d, 1\rangle), \iota(\langle e, 0\rangle)\big) \notin r^{\mathcal{J}^{\mathsf{L}}}$ with $\mathcal{J} = \iota(\mathcal{I}_0 \oplus \mathcal{Y}_0)$. Hence, $\big(d, e\big) \notin r^{\mathcal{J}^{\mathsf{L}}}$. Since $\mathcal{J}^{\mathsf{L}} \in \mathcal{W}_n$, this is a contradiction to the assumption $(d, e) \in (\mathsf{K}r)^{\mathcal{W}_n}$ and $d \notin \mathsf{Ind}$. Using symmetric arguments it can be shown that also the assumption $e \notin \Delta \setminus \mathsf{Ind}$ leads to a contradiction.

□

Thus, the extension of $\mathsf{K}C$ with an $\mathcal{ALC}$-concept C under a knowledge state that evolves from the epistemic model of an $\mathcal{ALC}$-KB by execution of primitive ground action sequence is either the whole domain or a finite set of named elements. Note that for this property to hold the restriction to a nominal-free and K-free concept C is necessary as the following example shows.

Example 26. Consider the following KB, concept and primitive ground action:

$$\mathcal{K}_0 = (\mathcal{T}_0 = \emptyset, \mathcal{A}_0 = \{\forall r.\neg A(b)\}), C = \forall r.(\neg A \sqcup \neg\{a\}) \text{ and } \alpha = (\mathsf{eff} : \{\neg A(a)\}, \mathsf{sense} : \emptyset).$$

It holds that

$$\mathcal{K}_0 \not\models \top \sqsubseteq C \text{ and } \big(\mathsf{K}C\big)^{\mathcal{M}(\mathcal{K}_0)} = \{b\}.$$

After executing α in $(\mathcal{I}, \mathcal{M}(\mathcal{K}_0)) \Longrightarrow_\alpha (\mathcal{I}', \mathcal{W}')$ we get $\big(\mathsf{K}C\big)^{\mathcal{W}'} = \Delta$. ▲

In the following we use Lemma 18 and Lemma 22.2 to equivalently rewrite epistemic concepts into objective concepts using nominals.

We introduce the notion of an instance function. The intuition is that the instance function captures known instances of concepts and known role successors of an individual represented as a set of nominal concepts.

Definition 27. Let Ind be a finite set of individuals. An *instance function w.r.t.* Ind maps concepts of the form $\mathbf{K}D$ to a subset of

$$\{\{a\} \mid a \in \mathsf{Ind}\} \cup \{\neg N\} \text{ with } N := \bigsqcup_{a \in \mathsf{Ind}} \{a\}$$

and an individual from Ind and a role name to a subset of $\{\{a\} \mid a \in \mathsf{Ind}\}$. Let $\mathcal{W}$ a set of interpretations. The *instance function of* $\mathcal{W}$ *w.r.t.* Ind, denoted by $\kappa_{\mathcal{W},\mathsf{Ind}}$, is defined as follows.

$$\begin{aligned}
&\kappa_{\mathcal{W},\mathsf{Ind}}(\mathbf{K}D) := \{\{a\} \mid a \in (\mathbf{K}D)^{\mathcal{W}}, a \in \mathsf{Ind}\} \cup \{\neg N \mid \exists c : c \in \Delta \setminus \mathsf{Ind}, c \in (\mathbf{K}D)^{\mathcal{W}}\};\\
&\kappa_{\mathcal{W},\mathsf{Ind}}(a, r) := \{\{b\} \mid (a, b) \in (\mathbf{K}r)^{\mathcal{W}}, b \in \mathsf{Ind}\}
\end{aligned}$$

for all concepts of the form $\mathbf{K}D$, $a \in \mathsf{Ind}$ and $r \in N_R$. Since the set of individuals Ind is fixed, we write only $\kappa_{\mathcal{W}}$ to denote an instance function of $\mathcal{W}$. ▲

Since we only consider non-empty knowledge states in the following, we assume for the rest of the report, that all simple $\mathcal{ALCOK}$-roles are of the form r or $\mathbf{K}r$ for some $r \in N_R$ and a generalized role has one of the following forms: r, $\neg r$, $\mathbf{K}r$, $\neg\mathbf{K}r$, $\mathbf{K}\neg r$, $\neg\mathbf{K}\neg r$ for some $r \in N_R$.

We define an operator $[\![\cdot,\cdot]\!]$ that rewrites subjective subconcepts and roles to objective ones based on a given instance function. Given an $\mathcal{ALCOK}$-concept D and an instance function κ an objective concept $[\![D, \kappa]\!]$ is defined by induction on the structure of D as shown in Figure 4.

$$\begin{aligned}
&[\![X, \kappa]\!] &&:= X \text{ with } X \in N_C \cup \{\top, \bot\} \text{ or } X = \{a\} \text{ for some } a \in \mathsf{Ind}\\
&[\![\neg D, \kappa]\!] &&:= \neg[\![D, \kappa]\!]\\
&[\![D_1 \sqcap D_2, \kappa]\!] &&:= [\![D_1, \kappa]\!] \sqcap [\![D_2, \kappa]\!]\\
&[\![D_1 \sqcup D_2, \kappa]\!] &&:= [\![D_1, \kappa]\!] \sqcup [\![D_2, \kappa]\!]\\
&[\![\exists r.E, \kappa]\!] &&:= \exists r.[\![E, \kappa]\!]\\
&[\![\forall r.E, \kappa]\!] &&:= \forall r.[\![E, \kappa]\!]\\
&[\![\mathbf{K}D, \kappa]\!] &&:= \bigsqcup \kappa(\mathbf{K}[\![D, \kappa]\!])\\
&[\![\exists\mathbf{K}r.D, \kappa]\!] &&:= \bigsqcup_{a \in \mathsf{Ind}} \left(\{a\} \sqcap \exists r.\left(\left(\bigsqcup \kappa(a, r)\right) \sqcap [\![D, \kappa]\!]\right)\right)\\
&[\![\forall\mathbf{K}r.D, \kappa]\!] &&:= \neg[\![\exists\mathbf{K}r.\neg D, \kappa]\!].
\end{aligned}$$

Figure 4: Operator for rewriting $\mathbf{K}$ using an instance function

Lemma 28. *Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ an interpretation, $\sigma = \alpha_0, \ldots, \alpha_{n-1}$ a sequence of primitive ground actions, C an $\mathcal{ALCOK}$-concept and $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation with $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$. It holds that $C^{\mathcal{J},\mathcal{W}_n} = [\![C, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}}$ for any interpretation $\mathcal{J}$.*

Proof. It follows from the definition of $[\![\cdot,\cdot]\!]$ that $[\![C, \kappa_{\mathcal{W}_n}]\!]$ is objective. Therefore $[\![C, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}}$ is well-defined. Note that the operator $[\![\cdot,\cdot]\!]$ implies the instance function only to concepts $\mathbf{K}D$ where D is objective. We show the claim by induction on the structure of C. The claim trivially holds if C is of the form A with $A \in N_C$, $\{a\}$ with $a \in \mathsf{Ind}$ or $\top$ or $\bot$.

$C = \neg D :$ Let $\mathcal{I}$ be an interpretation and assume by induction $D^{\mathcal{I},\mathcal{W}_n} = [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$:

$$(\neg D)^{\mathcal{I},\mathcal{W}_n} = \Delta \setminus D^{\mathcal{I},\mathcal{W}_n} = \Delta \setminus [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}} = (\neg[\![D, \kappa_{\mathcal{W}_n}]\!])^{\mathcal{I}} = [\![\neg D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}.$$

$C = \mathsf{K}D :$ $\Rightarrow$: Let $d \in (\mathsf{K}D)^{\mathcal{I},\mathcal{W}_n}$ for an interpretation $\mathcal{I}$ and $d \in \Delta$. We show $d \in [\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$. It holds that $d \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} D^{\mathcal{J},\mathcal{W}_n}$. Since by induction we have $D^{\mathcal{J},\mathcal{W}_n} = [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}}$ for any interpretation $\mathcal{J}$, it is implied that $d \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}}$. First, assume $d \in \mathsf{Ind}$. By definition of the instance function it follows that $\{d\} \in \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!])$. Therefore $\{d\}$ is a disjunct in $[\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!] = \bigsqcup \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!])$ and it follows that $d \in [\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$.
Now assume $d \in \Delta \setminus \mathsf{Ind}$. It is implied that $\neg N \in \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!])$. Therefore $\neg N$ is a disjunct in $[\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!]$ with $d \in (\neg N)^{\mathcal{I}}$. Consequently, $d \in [\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$.

$\Leftarrow$: Let $d \in [\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$ for an interpretation $\mathcal{I}$ and $d \in \Delta$. We have

$$[\![\mathsf{K}D, \kappa_{\mathcal{W}_n}]\!] = \bigsqcup \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!]) \text{ with } \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!]) \subseteq \{\{a\} \mid a \in \mathsf{Ind}\} \cup \{\neg N\}.$$

First, assume $d \in \mathsf{Ind}$. Since $d \notin (\neg N)^{\mathcal{I}}$, it follows that $\{d\} \in \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!])$ and therefore

$$d \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}} = \bigcap_{\mathcal{J}\in\mathcal{W}_n} D^{\mathcal{J},\mathcal{W}_n}$$

by induction. Consequently, $d \in (\mathsf{K}D)^{\mathcal{I},\mathcal{W}_n}$.
Now, assume $d \in \Delta \setminus \mathsf{Ind}$. It follows that $\neg N \in \kappa_{\mathcal{W}_n}(\mathsf{K}[\![D, \kappa_{\mathcal{W}_n}]\!])$. By definition of $\kappa_{\mathcal{W}_n}$ it follows that there exists a $c \in \Delta \setminus \mathsf{Ind}$ such that

$$c \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{J}} = \bigcap_{\mathcal{J}\in\mathcal{W}_n} D^{\mathcal{J},\mathcal{W}_n}$$

by induction. With $c \in \Delta \setminus \mathsf{Ind}$ and $c \in (\mathsf{K}D)^{\mathcal{W}_n}$ using Lemma 18 it follows that $\Delta \setminus \mathsf{Ind} \subseteq (\mathsf{K}D)^{\mathcal{W}_n}$. Hence, $d \in (\mathsf{K}D)^{\mathcal{W}_n}$.

$C = \exists\mathsf{K}r.D :$ Let $d \in (\exists\mathsf{K}r.D)^{\mathcal{I},\mathcal{W}_n}$ for some interpretation $\mathcal{I}$ and $d \in \Delta$

iff there exists an $e \in D^{\mathcal{I},\mathcal{W}_n}$ and $(d, e) \in \mathsf{K}r^{\mathcal{I},\mathcal{W}_n}$

iff there exists an $e \in [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$ (by induction) and $(d, e) \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} r^{\mathcal{J}}$

iff there exists an $e \in [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$, $(d, e) \in \bigcap_{\mathcal{J}\in\mathcal{W}_n} r^{\mathcal{J}}$ and $d, e \in \mathsf{Ind}$ by Lemma 22.2

iff there exists an $e \in [\![D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$, $d \in \mathsf{Ind}$ and $\{e\} \in \kappa_{\mathcal{W}_n}(d, r)$

iff $d \in [\![\exists\mathsf{K}r.D, \kappa_{\mathcal{W}_n}]\!]^{\mathcal{I}}$.

We omit the remaining cases. They can be proven using the induction hypothesis and the semantics of concepts. □

To compute the objective concept $[\![C, \kappa]\!]$ for a given $\mathcal{ALCOK}$-concept C and instance function κ, we need to compute the image of κ for the epistemic sub-concepts and roles of C.

Since the instance function is applied only to concepts $\mathsf{K}D$ where D is objective, we only need to determine the known instances of objective concepts after executing the sequence of ground actions σ. To do this we adapt the reduction approach from [BLM+05]. We construct a KB to capture the models contained in $\mathcal{W}_n$. For now we just ignore sensing.

Consider an initial $\mathcal{ALC}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and an action sequence $\sigma = \alpha_0\alpha_1 \dots \alpha_{n-1}$ with $\alpha_i = (\mathsf{eff}_i, \mathsf{sense}_i)$, $i = 0, \dots, n-1$. We say that the sequence $\mathcal{I}_0, \dots, \mathcal{I}_n$ of interpretations is *generated from* $\mathcal{I}_0$ *by* σ iff $\mathcal{I}_{i+1} = \mathcal{I}_i^{\mathcal{E}(\mathsf{eff}_i, \mathcal{I}_i)}$ for all $i = 0, \dots, n-1$. We now construct an $\mathcal{ALCO}$-KB $\mathcal{K}^\sigma_{\text{red}}$ such that any sequence of interpretations generated from an interpretation $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ by σ is encoded in a single interpretation that satisfies $\mathcal{K}^\sigma_{\text{red}}$ and the other way round, each model of $\mathcal{K}^\sigma_{\text{red}}$ encodes a sequence of interpretations that is generated from a model of $\mathcal{K}$ by σ. We recall the construction of $\mathcal{K}^\sigma_{\text{red}}$ and the pertinent results from [FMC+05]:

As before, $\mathcal{R} \subset N_C \cup N_R$ denotes a fixed finite set of relevant names that contains all names used in the input, i.e. in the initial KB, the action sequence and the projection query.

For each concept name $A \in \mathcal{R}$ and role name $r \in \mathcal{R}$ we introduce time-stamped copies $A^{(i)}$ and $r^{(i)}$ for all $i = 0, \dots, n$. Furthermore, for each subconcept C occurring in $\mathcal{K}$ or σ and for each i we use a fresh concept name $T_C^{(i)}$. The TBox $\mathcal{T}_{\text{sub}}$ consists of concept definitions according to Figure 5, i.e. exactly one axiom for each new name $T_C^{(i)}$. The reduction TBox $\mathcal{T}^\sigma_{\text{red}}$ is defined as follows:

$$\mathcal{T}^\sigma_{\text{red}} := \mathcal{T}_{\text{sub}} \cup \{T_C^{(0)} \sqsubseteq T_D^{(0)} \mid C \sqsubseteq D \in \mathcal{T}\} \cup \{N \equiv \bigsqcup_{a \in \mathsf{Ind}} \{a\}\}$$

$$\begin{aligned}
T_A^{(i)} &\equiv (N \sqcap A^{(i)}) \sqcup (\neg N \sqcap A^{(0)}) \\
T_{\{a\}}^{(i)} &\equiv \{a\} \\
T_{\neg C}^{(i)} &\equiv \neg T_C^{(i)} \\
T_{C \sqcap D}^{(i)} &\equiv T_C^{(i)} \sqcap T_D^{(i)} \\
T_{C \sqcup D}^{(i)} &\equiv T_C^{(i)} \sqcup T_D^{(i)} \\
T_{\exists r.C}^{(i)} &\equiv \Big(N \sqcap \big((\exists r^{(0)}.(\neg N \sqcap T_C^{(i)})) \sqcup (\exists r^{(i)}.(N \sqcap T_C^{(i)}))\big)\Big) \sqcup (\neg N \sqcap \exists r^{(0)}.T_C^{(i)}) \\
T_{\forall r.C}^{(i)} &\equiv \Big(N \rightarrow \big((\forall r^{(0)}.(\neg N \rightarrow T_C^{(i)})) \sqcap (\forall r^{(i)}.(N \rightarrow T_C^{(i)}))\big)\Big) \sqcap (\neg N \rightarrow \forall r^{(0)}.T_C^{(i)})
\end{aligned}$$

Figure 5: concept definitions in $\mathcal{T}_{\text{sub}}$ ([BLM+05])

To construct the reduction ABox we use a fresh auxiliary individual name $a_{\text{help}} \in \Delta \setminus \mathsf{Ind}$ and we introduce a new role name r_b for each $b \in \mathsf{Ind}$. The new role names are constrained as follows:

$$\mathcal{A}_{\text{aux}} := \{\big(\exists r_b.\{b\} \sqcap \forall r_b.\{b\}\big)(a) \mid a \in \mathsf{Ind} \cup \{a_{\text{help}}\}, b \in \mathsf{Ind}\}.$$

For a given $i \in \{0, \dots, n\}$ we translate an $\mathcal{ALCO}$-ground formula occurring in the input into a concept as follows:

$$\begin{aligned}
\mathsf{tcon}_i(C(a)) &:= \forall r_a.T_C^{(i)} \\
\mathsf{tcon}_i(r(a,b)) &:= \forall r_a.\exists r^{(i)}.\{b\} \\
\mathsf{tcon}_i(\neg\psi) &:= \neg\mathsf{tcon}_i(\psi) \\
\mathsf{tcon}_i(\psi_1 \vee \psi_2) &:= \mathsf{tcon}_i(\psi_1) \sqcup \mathsf{tcon}_i(\psi_2) \\
\mathsf{tcon}_i(\psi_1 \wedge \psi_2) &:= \mathsf{tcon}_i(\psi_1) \sqcap \mathsf{tcon}_i(\psi_2).
\end{aligned}$$

We use the following abbreviation for a ground formula ψ:

$$\psi^{(i)} := \mathsf{tcon}_i(\psi)(a_{\text{help}}).$$

For each $i \in \{1, \ldots, n\}$ the following ABoxes capture the action effects:

$$\mathcal{A}_{\text{eff}}^{(i)} := \{\big(\mathsf{tcon}_{i-1}(\psi) \to \mathsf{tcon}_i(\gamma)\big)(a_{\text{help}}) \mid \psi/\gamma \in \mathsf{eff}_i\}$$

and the ABox $\mathcal{A}_{\text{min}}^{(i)}$ contains

- for each $a \in \mathsf{Ind}$ and each concept name $A \in \mathcal{R}$ the following assertions

$$\Big(\big(A^{(i-1)} \sqcap \bigsqcap_{\psi/\neg A(a) \in \mathsf{eff}_i} \neg\mathsf{tcon}_{i-1}(\psi)\big) \to A^{(i)}\Big)(a)$$

$$\Big(\big(\neg A^{(i-1)} \sqcap \bigsqcap_{\psi/A(a) \in \mathsf{eff}_i} \neg\mathsf{tcon}_{i-1}(\psi)\big) \to \neg A^{(i)}\Big)(a)$$

- similarly for each pair of named elements $a, b \in \mathsf{Ind}$ and every role name $r \in \mathcal{R}$:

$$\Big(\big(\exists r^{(i-1)}.\{b\} \sqcap \bigsqcap_{\psi/\neg r(a,b) \in \mathsf{eff}_i} \neg\mathsf{tcon}_{i-1}(\psi)\big) \to \exists r^{(i)}.\{b\}\Big)(a)$$

$$\Big(\big(\forall r^{(i-1)}.\neg\{b\} \sqcap \bigsqcap_{\psi/r(a,b) \in \mathsf{eff}_i} \neg\mathsf{tcon}_{i-1}(\psi)\big) \to \forall r^{(i)}.\neg\{b\}\Big)(a).$$

Next, we encode the initial ABox:

$$\mathcal{A}_{\text{ini}} := \{\varphi^{(0)} \mid \varphi \in \mathcal{A}\}.$$

The reduction ABox as a whole is now obtained as follows:

$$\mathcal{A}_{\text{red}}^{\sigma} := \mathcal{A}_{\text{ini}} \cup \mathcal{A}_{\text{aux}} \cup \mathcal{A}_{\text{eff}}^{(1)} \cup \cdots \cup \mathcal{A}_{\text{eff}}^{(n)} \cup \mathcal{A}_{\text{min}}^{(1)} \cup \cdots \cup \mathcal{A}_{\text{min}}^{(n)}.$$

The reduction KB is given by $\mathcal{K}_{\text{red}}^{\sigma} = (\mathcal{T}_{\text{red}}^{\sigma}, \mathcal{A}_{\text{red}}^{\sigma})$.

In addition we use the following auxiliary notions: For a given objective concept, axiom or formula X and action sequence σ of length n the TBox, denoted by $\mathcal{T}_{\mathsf{sub}(X)}^{\sigma}$, consists of a definition for each new concept name $T_D^{(i)}$ with $D \in \mathsf{sub}(X)$ and $i \in \{0, \ldots, n\}$ according to Figure 5. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a TBox $\mathcal{T}'$ and an ABox $\mathcal{A}'$, we write $\mathcal{K} \cup \mathcal{T}'$ to denote the KB $(\mathcal{T} \cup \mathcal{T}', \mathcal{A})$ and $\mathcal{K} \cup \mathcal{A}'$ for the KB $(\mathcal{T}, \mathcal{A} \cup \mathcal{A}')$.

In the next lemma we show that the models of $\mathcal{K}_{\text{red}}^{\sigma}$ correctly encode the sequences of interpretations generated from models of $\mathcal{K}$ by σ.

Lemma 29. *Let $\mathcal{K}$ be an $\mathcal{ALC}$-KB, $\sigma = \alpha_0\alpha_1 \ldots \alpha_{n-1}$ a sequence of ground actions, $\mathcal{K}_{red}^{\sigma}$ the reduction KB as constructed above, ψ an $\mathcal{ALCO}$-ground formula and C an $\mathcal{ALCO}$-concept with $\mathsf{sub}(\psi) \cup \mathsf{sub}(C) \subseteq \mathsf{sub}(\mathcal{K})$.*

1. *For every sequence of interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ generated from an interpretation $\mathcal{I}_0$ with $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{K}_{red}^{\sigma}$ and for every $j \in \{0, \ldots, n\}$ it holds that*

   (a) $\mathcal{I}_j \models \psi$ iff $\mathcal{J} \models \psi^{(j)}$ and

   (b) $C^{\mathcal{I}_j} = (T_C^{(j)})^{\mathcal{J}}$.

2. *For every interpretation $\mathcal{J}$ with $\mathcal{J} \models \mathcal{K}_{red}^{\sigma}$, there exists $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ such that for the sequence $\mathcal{I}_0, \ldots, \mathcal{I}_n$ generated from $\mathcal{I}_0$ by σ it holds that*

(a) $\mathcal{I}_j \models \psi$ *iff* $\mathcal{J} \models \psi^{(j)}$ *and*

(b) $C^{\mathcal{I}_j} = (T_C^{(j)})^{\mathcal{J}}$ *for all* $j \in \{0, \ldots, n\}$.

Proof. We omit the proof here. Essentially, the proof works in the same way as the one given in [FMC+05] (page 17, Theorem 14). □

Next, we incorporate the sensing results of an action sequence.

Definition 30. Let $\sigma = \alpha_0\alpha_1 \ldots \alpha_{n-1}$ with $\alpha_i = (\mathsf{eff}_i, \mathsf{sense}_i)$ and $i = 0, \ldots, n-1$ be a sequence of primitive ground actions. A *sensing result of* σ is a sequence $\mathfrak{A} = \mathsf{S}_0\mathsf{S}_1 \cdots \mathsf{S}_{n-1}$ such that for all $i \in \{0, \ldots, n-1\}$ it holds that $\mathsf{S}_i \subseteq \mathsf{sense}_i \cup \neg\mathsf{sense}_i$ and $\psi \in \mathsf{S}_i$ iff $\neg\psi \notin \mathsf{S}_i$ for all $\psi \in \mathsf{sense}_i \cup \neg\mathsf{sense}_i$. Let $\mathcal{I}_0$ be an interpretation and $\mathcal{I}_0, \ldots, \mathcal{I}_n$ the sequence of interpretations generated from $\mathcal{I}_0$ by σ. The *sensing result of* σ *w.r.t.* $\mathcal{I}_0$, denoted by $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathsf{S}_0\mathsf{S}_1 \cdots \mathsf{S}_{n-1}$, is given by $\mathsf{S}_i = \{\psi \mid \psi \in \mathsf{sense}_i \cup \neg\mathsf{sense}_i, \mathcal{I}_i \models \psi\}$ for all $i \in \{0, \ldots, n-1\}$. ▲

The reduction ABox $\mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ for a given sensing result $\mathfrak{A}$ of σ is given by

$$\mathcal{A}^{\mathfrak{A}}_{\text{sense}} := \bigcup_{i \in \{0,\ldots,n-1\}} \{\psi^{(i)} \mid \psi \in \mathsf{S}_i\}.$$

Lemma 31. *Let* $\mathcal{K}$ *and* σ *be as above and* $\mathfrak{A} = \mathsf{S}_0\mathsf{S}_1 \cdots \mathsf{S}_{n-1}$ *a sensing result of* σ*. There exists an* $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ *such that* $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathfrak{A}$ *iff* $\mathcal{K}^{\sigma}_{red} \cup \mathcal{A}^{\mathfrak{A}}_{sense}$ *is consistent.*

Proof.

⇒: Assume there exists an $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ such that $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathfrak{A}$. Let $\mathcal{I}_0, \ldots, \mathcal{I}_n$ be the sequence generated from $\mathcal{I}_0$ by σ. For all $i \in \{0, \ldots, n-1\}$ and all $\psi \in \mathsf{S}_i$ it holds by assumption that $\mathcal{I}_i \models \psi$. From Lemma 29.1 it follows that there exists a model $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{K}^{\sigma}_{\text{red}}$ and $\mathcal{J} \models \psi^{(i)}$ for all $i \in \{0, \ldots, n-1\}$ and all $\psi \in \mathsf{S}_i$. This implies $\mathcal{J} \models \mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$.

⇐: Assume $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent. There exists a model $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$. It follows from Lemma 29.2 that there exists $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ such that for the sequence $\mathcal{I}_0, \ldots, \mathcal{I}_n$ generated from $\mathcal{I}_0$ by σ it holds that $\mathcal{J} \models \psi^{(i)}$ iff $\mathcal{I}_i \models \psi$ for an $\mathcal{ALCO}$-ground formula and all $i \in \{0, \ldots, n\}$. Since $\mathcal{J} \models \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$, it follows that $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathfrak{A}$.

□

Using the reduction ABox for the sensing results we can now represent the knowledge state after a sequence of ground actions was performed.

Lemma 32. *Let* $\mathcal{K}$*,* σ*,* ψ*,* C *and* $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ *be as above and* $(\mathcal{I}_n, \mathcal{W}_n)$ *the epistemic interpretation with* $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma} (\mathcal{I}_n, \mathcal{W}_n)$ *and* $\mathfrak{A} = \mathfrak{A}^{\mathcal{I}_0,\sigma}$.

1. *For every interpretation* $\mathcal{J}_n \in \mathcal{W}_n$ *there exists an interpretation* $\mathcal{J}$ *such that* $\mathcal{J} \models \mathcal{K}^{\sigma}_{red} \cup \mathcal{A}^{\mathfrak{A}}_{sense}$ *and it holds that (a)* $\mathcal{J}_n \models \psi$ *iff* $\mathcal{J} \models \psi^{(n)}$ *and (b)* $C^{\mathcal{J}_n} = (T_C^{(n)})^{\mathcal{J}}$.

2. *For every interpretation* $\mathcal{J}$ *with* $\mathcal{J} \models \mathcal{K}^{\sigma}_{red} \cup \mathcal{A}^{\mathfrak{A}}_{sense}$ *there exists an interpretation* $\mathcal{J}_n \in \mathcal{W}_n$ *such that (a)* $\mathcal{J}_n \models \psi$ *iff* $\mathcal{J} \models \psi^{(n)}$ *and (b)* $C^{\mathcal{J}_n} = (T_C^{(n)})^{\mathcal{J}}$.

Proof.

1. Let $\mathcal{J}_n \in \mathcal{W}_n$ with

$$(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\alpha_0} (\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\alpha_1} \cdots \Longrightarrow_{\alpha_{n-1}} (\mathcal{I}_n, \mathcal{W}_n).$$

   There exists $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ such that $\mathcal{J}_0, \dots, \mathcal{J}_n$ is the sequence generated from $\mathcal{J}_0$ by $\sigma = \alpha_0\alpha_1 \cdots \alpha_{n-1}$. By the definition of the action semantics it follows that $\mathcal{J}_i \sim_{\alpha_i} \mathcal{I}_i$ for all $i \in \{0, \dots, n-1\}$. Therefore, $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathfrak{A}^{\mathcal{J}_0,\sigma}$. By Lemma 29.1 there exists an interpretation $\mathcal{J}$ with $\mathcal{J} \models \mathcal{K}^{\sigma}_{\text{red}}$ such that $\mathcal{J} \models \mathcal{A}^{\mathfrak{A}^{\mathcal{J}_0,\sigma}}_{\text{sense}} = \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ and $\mathcal{J}_n \models \psi$ iff $\mathcal{J} \models \psi^{(n)}$ and $C^{\mathcal{J}_n} = (T^{(n)}_C)^{\mathcal{J}}$ for an $\mathcal{ALCO}$-ground formula ψ and an $\mathcal{ALCO}$-concept C.

2. Let $\mathcal{J}$ be an interpretation such that $\mathcal{J} \models \mathcal{K}^{\sigma}_{\text{red}}$ and $\mathcal{J} \models \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$. According to Lemma 29.2, there exists $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ such that the interpretations in the sequence $\mathcal{J}_0, \dots, \mathcal{J}_n$ generated from $\mathcal{J}_0$ by σ satisfy the claims (a) and (b) in Lemma 29.2. It remains to be shown that $\mathcal{J}_n \in \mathcal{W}_n$. Since $\mathcal{J} \models \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ and Lemma 29.2 holds, it follows that $\mathfrak{A}^{\mathcal{J}_0,\sigma} = \mathfrak{A}$. Therefore, $\mathcal{J}_i \sim_{\alpha_i} \mathcal{I}_i$ for all $i \in \{0, \dots, n-1\}$ and consequently, $\mathcal{J}_n \in \mathcal{W}_n$.

□

Based on the reduction KB and the representation of a sensing result we can now compute the image of the corresponding instance function. We choose an individual name $c_u \in \Delta \setminus \mathsf{Ind}$ not occurring in $\mathcal{K}$ or σ.

Given an initial KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a sequence of ground action σ of length n and a sensing result $\mathfrak{A}$ of σ such that the KB $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent, the instance function $\kappa_{\mathfrak{A}}$ is defined for concepts of the form $\mathbf{K}C$ where C is objective as follows:

$$\begin{aligned}\kappa_{\mathfrak{A}}(\mathbf{K}C) := \{&\{a\} \mid \mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}(C)} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models T^{(n)}_C(a), a \in \mathsf{Ind}\} \cup \\ &\{\neg N \mid \mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}(C)} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models T^{(n)}_C(c_u)\}\end{aligned}$$

For concepts $\mathbf{K}D$ where D is not objective we define $\kappa_{\mathfrak{A}}(\mathbf{K}D) := \emptyset$. And for an individual a and role name r $\kappa_{\mathfrak{A}}$ is defined as follows:

$$\kappa_{\mathfrak{A}}(a, r) := \{b \mid \mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}(C)} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models r^{(n)}(a, b), b \in \mathsf{Ind}\}.$$

Lemma 33. *Let $\mathcal{K}$ and σ be as above, D an $\mathcal{ALCOK}$-concept, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$, $\mathfrak{A} = \mathfrak{A}^{\mathcal{I}_0,\sigma}$ the sensing result of σ w.r.t. $\mathcal{I}_0$ and $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation with $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma} (\mathcal{I}_n, \mathcal{W}_n)$. It holds that $[\![D, \kappa_{\mathfrak{A}}]\!] = [\![D, \kappa_{\mathcal{W}_n}]\!]$.*

Proof. The claim is a direct consequence of Lemma 32. □

To check whether a subjective projection query ψ is valid after executing σ in $\mathcal{K}$ we proceed as follows: for all sensing results $\mathfrak{A}$ of σ such that $\mathcal{K}^{\sigma}_{\text{red}}$ is consistent with $\mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ we use the operator $[\![\cdot, \kappa_{\mathfrak{A}}]\!]$ to compute an equivalent objective query $\widehat{\psi}$ and then check whether the transformed projection query $\widehat{\psi}^{(n)}$ is entailed by $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$.

We can assume w.l.o.g. that ψ consists only of concept assertions. Role assertions can be equivalently replaced by concept assertions:

$$\begin{aligned}(\neg)r(a, b) &\rightsquigarrow (\neg)\exists r.\{b\}(a) \\ (\neg)\mathbf{K}r(a, b) &\rightsquigarrow (\neg)\exists \mathbf{K}r.\{b\}(a) \\ (\neg)\mathbf{K}\neg r(a, b) &\rightsquigarrow (\neg)(\mathbf{K}\forall r.\neg\{b\})(a).\end{aligned}$$

$\widehat{\psi}_{\mathfrak{A}}$ denotes the formula that is obtained from ψ by replacing each concept assertion $C(a)$ in ψ by the assertion $[\![C, \kappa_{\mathfrak{A}}]\!](a)$. Clearly, $\widehat{\psi}_{\mathfrak{A}}$ is objective.

The following lemma is a direct consequence of Lemma 28 and 33.

Lemma 34. *Let $\mathcal{K}$ and σ be as above, ψ an $\mathcal{ALCOK}$-ground formula, $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$, $\mathfrak{A} = \mathfrak{A}^{\mathcal{I}_0,\sigma}$ and $(\mathcal{I}_n, \mathcal{W}_n)$ the epistemic interpretation with $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$. It holds that $(\mathcal{I}_n, \mathcal{W}_n) \mid\models \psi$ iff $\mathcal{I}_n \models \widehat{\psi}_{\mathfrak{A}}$.*

Finally, the projection problem can be decided by a finite number of non-epistemic entailment checks.

Lemma 35. *Let $\mathcal{K}$ be a consistent $\mathcal{ALC}$-KB, σ a sequence of primitive ground actions and ψ an $\mathcal{ALCOK}$-ground formula. It holds that ψ is valid after executing σ in $\mathcal{K}$ iff for all sensing results $\mathfrak{A}$ of σ such that $\mathcal{K}^\sigma_{red} \cup \mathcal{A}^{\mathfrak{A}}_{sense}$ is consistent it holds that $\mathcal{K}^\sigma_{red} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{sense} \models \widehat{\psi}^{(n)}_{\mathfrak{A}}$.*

Proof.

$\Rightarrow$: Assume to the contrary that ψ is valid after executing σ in $\mathcal{K}$ and there exists a sensing result $\mathfrak{A}$ of σ such that $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent and $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \not\models \widehat{\psi}^{(n)}_{\mathfrak{A}}$. Since $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent, there exists $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ such that $\mathfrak{A}^{\mathcal{I}_0,\sigma} = \mathfrak{A}$ (by Lemma 31). Since $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \not\models \widehat{\psi}^{(n)}_{\mathfrak{A}}$ holds by assumption, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ and $\mathcal{J} \not\models \widehat{\psi}^{(n)}_{\mathfrak{A}}$. Let $(\mathcal{I}_n, \mathcal{W}_n)$ be the epistemic interpretation such that

$$(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\alpha_0} (\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\alpha_1} \cdots \Longrightarrow_{\alpha_{n-1}} (\mathcal{I}_n, \mathcal{W}_n).$$

with $\sigma = \alpha_0\alpha_1 \cdots \alpha_{n-1}$. By Lemma 32 there exists $\mathcal{J}_n \in \mathcal{W}_n$ such that $\mathcal{J}_n \not\models \widehat{\psi}_{\mathfrak{A}}$. Since $\mathcal{J}_n \in \mathcal{W}_n$, there exists (by definition of the action semantics) a sequence $\mathcal{J}_0, \ldots, \mathcal{J}_n$ generated from $\mathcal{J}_0$ by σ such that $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ and $\mathcal{I}_i \sim_{\alpha_i} \mathcal{J}_i$ for all $i = 0, \ldots, n-1$. Therefore and by the definition of the action semantics we obtain $(\mathcal{J}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{J}_n, \mathcal{W}_n)$. With $\mathcal{J}_n \not\models \widehat{\psi}_{\mathfrak{A}}$ and Lemma 34 it follows that $(\mathcal{J}_n, \mathcal{W}_n) \mid\not\models \psi$. Since $(\mathcal{J}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{J}_n, \mathcal{W}_n)$ this is a contradiction to the assumption that ψ is valid after executing σ in $\mathcal{K}$.

$\Leftarrow$: Assume to the contrary that ψ is *not* valid after executing σ in $\mathcal{K}$ and for all sensing results $\mathfrak{A}$ of σ such that $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent it holds that $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models \widehat{\psi}^{(n)}_{\mathfrak{A}}$. There exists $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ and an epistemic interpretation $(\mathcal{I}_n, \mathcal{W}_n)$ such that $(\mathcal{I}_0, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{I}_n, \mathcal{W}_n)$ and $(\mathcal{I}_n, \mathcal{W}_n) \mid\not\models \psi$. Let $\mathfrak{A} = \mathfrak{A}^{\mathcal{I}_0,\sigma}$. By Lemma 31 the KB $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent. From Lemma 34 and $(\mathcal{I}_n, \mathcal{W}_n) \mid\not\models \psi$ it follows that $\mathcal{I}_n \not\models \widehat{\psi}_{\mathfrak{A}}$. Since it holds that $\mathcal{I}_n \in \mathcal{W}_n$, it is implied by Lemma 32 that there exists a model $\mathcal{J}$ of $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ and $\mathcal{J} \not\models \widehat{\psi}^{(n)}_{\mathfrak{A}}$. Since $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent, it follows by assumption that $\mathcal{K}^\sigma_{\text{red}} \cup \mathcal{T}^\sigma_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models \widehat{\psi}^{(n)}_{\mathfrak{A}}$ which is a contradiction.

□

We can now show the main result of this section. The complexity of the decision procedure for the projection problem is measured in the size of the input, i.e. the number of symbols needed to write down the projection query ψ, all axioms in $\mathcal{K}$ and the action sequence σ.

Theorem 36. *Let $\mathcal{K}$ = ($\mathcal{T}$,$\mathcal{A}$) be an initial $\mathcal{ALC}$-KB, σ a sequence of primitive ground action and ψ an $\mathcal{ALCOK}$-ground formula. The problem of deciding whether or not ψ is valid after executing σ in $\mathcal{K}$ is* ExpTime*-complete and* PSpace*-complete if $\mathcal{T}$ is empty.*

Proof. First we show that the projection problem is decidable in ExpTime. The decision procedure consists of the following steps:

1. The KB $\mathcal{K}^{\sigma}_{\text{red}}$ is constructed. This can be done in polynomial time in the size of $\mathcal{K}$ and σ and $\mathcal{K}^{\sigma}_{\text{red}}$ is of polynomial size.
2. All sensing results $\mathfrak{A}$ of σ each of polynomial size in the size of σ are enumerated. There are exponentially many sensing results. They can be enumerated in exponential time.
3. For each sensing result $\mathfrak{A}$ we check whether $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent. Since $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is of polynomial size, the consistency check can be done in exponential time [BCM$^+$03].
4. For each $\mathfrak{A}$ such that $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ is consistent the objective formula $\widehat{\psi}_{\mathfrak{A}}$ is constructed. To do this we need to compute $[\![C, \kappa_{\mathfrak{A}}]\!]$ for all concepts C in ψ. In order to compute the concept $[\![C, \kappa_{\mathfrak{A}}]\!]$ for a given C, the instance function $\kappa_{\mathfrak{A}}$ is applied exactly m times, where m is the number of Ks occurring in C. The size of a concept $[\![D, \kappa_{\mathfrak{A}}]\!]$ is polynomial in the size of D and $|\mathsf{Ind}|$. The computation of the set $\kappa_{\mathfrak{A}}(\mathsf{K}[\![D, \kappa_{\mathfrak{A}}]\!])$ requires $|\mathsf{Ind}| + 1$ many instance checks w.r.t. the KB $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}([\![D,\kappa_{\mathfrak{A}}]\!])} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$. Each instance check can be done in exponential time. The computation of $\kappa_{\mathfrak{A}}(a, r)$ requires $|\mathsf{Ind}| \cdot |\mathsf{Ind}|$ many instance checks in exponential time. Thus $\widehat{\psi}_{\mathfrak{A}}$ is of polynomial size and can be computed in exponential time.
5. Finally, we check whether $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \models \widehat{\psi}^{(n)}_{\mathfrak{A}}$ for each consistent sensing result. Each of the exponentially many checks can be done in exponential time.

In sum, the decision procedure requires at most exponential time in the size of the input. Therefore we get an ExpTime upper-bound for the complexity of the projection problem. The matching lower-bound follows from the ExpTime-hardness of ABox consistency in $\mathcal{ALC}$ w.r.t. TBoxes with CIs.

Next we consider the case where the initial TBox $\mathcal{T}$ is empty. We show that the satisfiability problem can be solved using only polynomial space in the size of the input. The first step is to guess a sensing result $\mathfrak{A}$ of σ. Since the KB $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}}$ only has an acyclic TBox, consistency is decidable in PSpace. The computation of $\widehat{\psi}_{\mathfrak{A}}$ requires a polynomial number of instance checks that are solved each in polynomial space. The check whether $\mathcal{K}^{\sigma}_{\text{red}} \cup \mathcal{T}^{\sigma}_{\mathsf{sub}(\widehat{\psi}_{\mathfrak{A}})} \cup \mathcal{A}^{\mathfrak{A}}_{\text{sense}} \cup \{\widehat{\psi}^{(n)}_{\mathfrak{A}}\}$ is consistent can be done in PSpace. Since NPSpace = PSpace, we obtain a PSpace upper-bound. The matching lower-bound is obtained as in the previous case. □

4 Verification of Knowledge-Based Programs

We define a programming language that allows us to build complex actions describing the behavior of a knowledge-based agent. To specify desired properties of such programs we use a temporal extension of $\mathcal{ALCOK}$. Our main objective is to identify fragments of the programming language such that the verification problem, i.e. the problem of deciding whether all runs of a program satisfy the specified property, is decidable.

4.1 Syntax and Semantics of $\mathcal{ALCOK}$-Golog Programs

In this section we define the syntax and semantics of a Golog-like action programming language that uses the action formalism as introduced in the previous section. Program expressions describe how a complex action is constructed from primitive actions using programming constructs and tests formulated in $\mathcal{ALCOK}$.

Definition 37 ($\mathcal{ALCOK}$-Golog program)**.** Let α be a primitive action and ψ an $\mathcal{ALCOK}$-formula. A *program expression* is defined inductively as follows.

- α is a program expression.
- The *test* $\psi?$ is a program expression.
- The *empty program* $[]$ is a program expression.
- If δ is a program expression, then the *non-deterministic iteration* of δ, denoted by $(\delta)^*$, is a program expression.
- If δ_1 and δ_2 are program expressions, then the *sequence* of δ_1 and δ_2, denoted by $(\delta_1; \delta_2)$, is a program expression.
- If δ_1 and δ_2 are program expressions, then the *non-deterministic choice* between δ_1 and δ_2, denoted by $(\delta_1|\delta_2)$, is a program expression.
- Let $n \geq 0$. If $x_1, ..., x_n$ is a sequence of variables, $\psi?$ a test and δ a program expression, then $(\mathsf{pick}(x_1, ..., x_n) : \psi?.\delta)$ (*non-deterministic choice of arguments*) is also a program expression. The expression $\mathsf{pick}(x_1, ..., x_n) : \psi?$ is called *guarded pick* and the test $\psi?$ is called *guard*.

The set of all variables occurring in a program expression is denoted by $\mathsf{Var}(\delta)$. The set of free variables, i.e. those that are not bound by a guarded pick, of a program expression δ, is denoted by $\mathsf{FVar}(\delta)$, given by

- $\mathsf{FVar}(\alpha) := \mathsf{Var}(\alpha)$,
- $\mathsf{FVar}(\psi?) := \mathsf{Var}(\psi)$;
- $\mathsf{FVar}([]) := \emptyset$,
- $\mathsf{FVar}((\delta)^*) := \mathsf{FVar}(\delta)$,
- $\mathsf{FVar}(\delta_1; \delta_2) := \mathsf{FVar}(\delta_1) \cup \mathsf{FVar}(\delta_2)$,
- $\mathsf{FVar}(\delta_1|\delta_2) := \mathsf{FVar}(\delta_1) \cup \mathsf{FVar}(\delta_2)$,
- $\mathsf{FVar}(\mathsf{pick}(x_1, ..., x_n) : \psi?.\delta) := (\mathsf{Var}(\psi) \cup \mathsf{FVar}(\delta)) \setminus \{x_1, ..., x_n\}$.

For a guarded pick of the form $\mathsf{pick}(x_1, ..., x_n) : \psi?$ we define

$$\mathsf{FVar}(\mathsf{pick}(x_1, ..., x_n) : \psi?) := \mathsf{Var}(\psi) \setminus \{x_1, \ldots, x_n\}.$$

A *closed* program expression has no free variables.

Let $\epsilon = (\mathsf{eff} : \{\mathit{Term}(p)\}, \mathsf{sense} : \emptyset)$ and $\mathfrak{f} = (\mathsf{eff} : \{\mathit{Fail}(p)\}, \mathsf{sense} : \emptyset)$ be two predefined actions for indicating termination and failure of a program, respectively.

An *$\mathcal{ALCOK}$-Golog program* $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ consists of an initial consistent $\mathcal{ALC}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a finite set of primitive actions and a closed program expression δ satisfying the following conditions:

- $\{\neg \mathit{Term}(p), \neg \mathit{Fail}(p)\} \subseteq \mathcal{A}$,
- the names *Term* and *Fail* do not occur in $\mathcal{T}$ and $\mathcal{A} \setminus \{\neg \mathit{Term}(p), \neg \mathit{Fail}(p)\}$,

- $\{\epsilon, \mathfrak{f}\} \subseteq \Sigma$ and
- every primitive action occurring in δ is contained in $\Sigma \setminus \{\epsilon, \mathfrak{f}\}$.

An $\mathcal{ALCOK}$-Golog program $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ is called a *knowledge-based program* if all tests occurring in δ are subjective. ▲

Together with the other constructs, tests can for example be used to express while-loops and if-then-else statements:

$$\textbf{while } \psi? \textbf{ do } \delta \textbf{ end} := (\psi?; \delta)^*; \neg\psi?$$
$$\textbf{if } \psi? \textbf{ then } \delta_1 \textbf{ else } \delta_2 \textbf{ end} := \psi?; \delta_1 \mid \neg\psi?; \delta_2$$

In the following we use the notation $\vec{x}$ to denote an n-tuple of variables $\vec{x} = (x_1, \ldots, x_n)$ with $n \in \mathbb{N}$. Likewise, $\vec{a}$ denotes an n-tuple of domain elements $(a_1, \ldots, a_n)$ with $n \in \mathbb{N}$ and $a_i \in \Delta$, $i = 1, \ldots, n$.

For the execution of a program we split up the program expression into its atomic programs and then execute these atomic programs step by step. An *atomic program*, denoted by $\mathfrak{a}$, is either a primitive action, a test or a guarded pick. As in [ZC14, BZ13a] we introduce two functions $\mathsf{head}(\cdot)$ and $\mathsf{tail}(\cdot, \cdot)$. Intuitively, $\mathsf{head}(\delta)$ contains those atomic programs that can be executed first when executing the program expression δ. For $\mathfrak{a} \in \mathsf{head}(\delta)$, $\mathsf{tail}(\mathfrak{a}, \delta)$ yields the remainder of the program, i.e., the part that still needs to be executed after $\mathfrak{a}$ has been executed. The functions $\mathsf{head}(\cdot)$ and $\mathsf{tail}(\cdot, \cdot)$ are defined by induction on the size of program expressions as given in Table 3.

Program expr. δ	$\mathsf{head}(\delta)$	$\mathsf{tail}(\mathfrak{a}, \delta)$ with $\mathfrak{a} \in \mathsf{head}(\delta)$
$[]$	$\{\epsilon\}$	$\{[]\}$
α	$\{\alpha\}$	$\{[]\}$
$\psi?$	$\{\psi?\}$	$\{[]\}$
δ^*	$\{\epsilon\} \cup \mathsf{head}(\delta)$	$\{\delta'; (\delta)^* \mid \delta' \in \mathsf{tail}(\mathfrak{a}, \delta), \mathfrak{a} \neq \epsilon\} \cup$ $\{[] \mid \mathfrak{a} = \epsilon\}$
$\delta_1; \delta_2$	$\{\mathfrak{a} \in \mathsf{head}(\delta_1) \mid \mathfrak{a} \neq \epsilon\} \cup$ $\{\mathfrak{a} \in \mathsf{head}(\delta_2) \mid \epsilon \in \mathsf{head}(\delta_1)\}$	$\{\delta'; \delta_2 \mid \delta' \in \mathsf{tail}(\mathfrak{a}, \delta_1)\} \cup$ $\{\delta' \in \mathsf{tail}(\mathfrak{a}, \delta_2) \mid \epsilon \in \mathsf{head}(\delta_1)\}$
$\delta_1 \mid \delta_2$	$\mathsf{head}(\delta_1) \cup \mathsf{head}(\delta_2)$	$\mathsf{tail}(\mathfrak{a}, \delta_1) \cup \mathsf{tail}(\mathfrak{a}, \delta_2)$
$\mathsf{pick}(\vec{x}) : \psi?.\delta$	$\{\mathsf{pick}(\vec{x}) : \psi?\}$	$\{\delta\}$

Table 3: Definition of head and tail of a program expression and atomic program, respectively. For $\mathfrak{a} \notin \mathsf{head}(\delta)$ we define $\mathsf{tail}(\mathfrak{a}, \delta) := \emptyset$.

Intuitively, executing a program δ means first execute an atomic program of its head, then an atomic program of the head of its tail, etc. Thus, the program δ is consumed step by step and if a program is "eaten up" completely, the termination action ϵ can be found in the head. We call a program expression that can be reached by a sequence of such head and tail applications a reachable subprogram.

Definition 38. Let δ be a program expression. The program expression ρ is a *reachable subprogram* of δ if there is an $n \geq 0$ and program expressions $\delta_0, \delta_1, \ldots, \delta_n$ such that $\delta_0 = \delta$, $\delta_n = \rho$, and for all $i = 0, \cdots, n-1$ there exists $\mathfrak{a}_i \in \mathsf{head}(\delta_i)$ such that $\delta_{i+1} \in \mathsf{tail}(\mathfrak{a}_i, \delta_i)$. We denote the set of all reachable subprograms of δ by $\mathsf{sub}(\delta)$. ▲

Clearly, the number of reachable subprograms of a given program expression i s bounded in the size of the program expression. The size $|\delta|$ of a program expression δ is given by the number of symbols needed to write δ.

Lemma 39. *Let δ be a program expression. The cardinality of $\mathsf{sub}(\delta)$ is polynomially bounded in the size $|\delta|$ of δ.*

Proof. The proof can be done with very similar arguments as used for the proof of Lemma 11 in [BZ13b] (page 9). We omit the proof here. □

Note that in [Cla14, ZC14, BZ13a] tests in program expressions are handled differently. They are viewed as conditions that are required to be satisfied before the next primitive action can be executed and do not cause separate execution steps in order to avoid unintended interleaving of tests and action in presence of a concurrency operator. Since we do not consider the interleaving constructor here, we use here a slightly simpler definition of head and tail.

Note that a reachable subprogram of a closed program expression may contain free variables. We show the following properties of closed program expressions.

Lemma 40. *Let δ be a closed program expression. The following properties are satisfied.*

1. *If $\mathfrak{a} \in \mathsf{head}(\delta)$, then $\mathsf{FVar}(\mathfrak{a}) = \emptyset$.*
2. *If $\mathfrak{a} \in \mathsf{head}(\delta)$ is a test or primitive action and $\rho \in \mathsf{tail}(\mathfrak{a}, \delta)$, then ρ is closed.*
3. *If $\mathfrak{a} \in \mathsf{head}(\delta)$, $\rho \in \mathsf{tail}(\mathfrak{a}, \delta)$ and $\mathfrak{a}$ is a guarded pick of the form $\mathsf{pick}(\vec{x}) : \psi?$, then $\mathsf{Var}(\psi) \subseteq \vec{x}$ and $\mathsf{FVar}(\rho) \subseteq \vec{x}$.*

Proof. The lemma can be proven by a simple induction on the structure of δ using the definitions of head and tail. □

Semantically, a *state* of a program is of the form $\langle(\mathcal{I}, \mathcal{W}), \delta\rangle$ where $(\mathcal{I}, \mathcal{W})$ is an epistemic interpretation and δ a closed program expression. We say that an *atomic program* $\mathfrak{a} \in \mathsf{head}(\delta)$ *is executable in* $\langle(\mathcal{I}, \mathcal{W}), \delta\rangle$ iff one of the following conditions is satisfied:

- $\mathfrak{a}$ is a primitive action;
- $\mathfrak{a}$ is a test of the form $\psi?$ and $(\mathcal{I}, \mathcal{W}) \models \psi$;
- $\mathfrak{a}$ is a guarded pick of the form $\mathsf{pick}(\vec{x}) : \psi?$ and there exists a variable mapping ν such that $(\mathcal{I}, \mathcal{W}) \models \psi^{\nu}$.

From Lemma 40 it follows that executability is well defined.

Intuitively, the agent executes a guarded pick by non-deterministically choosing a binding $\vec{a}$ from Δ for the $\vec{x}$ such that ψ with $\vec{x}$ replaced by $\vec{a}$ is satisfied, after which it executes δ using the same bindings.

Let δ be a program expression and ν a variable mapping. The closed program expression δ^{ν} is obtained from δ by simultaneously replacing each occurrence of a free variable $x \in \mathsf{FVar}(\delta)$ in δ by $\nu(x)$. The set of all instantiated reachable subprograms of a program expression δ is given by

$$\mathsf{subg}(\delta) := \{\zeta^{\nu} \mid \zeta \in \mathsf{sub}(\delta), \nu \text{ is a variable mapping}\}.$$

Let $\mathcal{K}$ be an objective KB and Σ a set of primitive actions. We define the set of all ground actions and all possible evolutions of epistemic interpretations, respectively, as follows.

$$\begin{aligned}
\Sigma_{\mathsf{g}} &:= \{\alpha^{\nu} \mid \alpha \in \Sigma, \nu \text{ is a variable mapping}\};\\
\mathfrak{I}_{\mathcal{K},\Sigma} &:= \{(\mathcal{I}', \mathcal{W}') \mid \text{ there is } \mathcal{I} \in \mathcal{M}(\mathcal{K}) \text{ and } \sigma \in \Sigma_{\mathsf{g}}^{*} \text{ s.t. } (\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_{\sigma} (\mathcal{I}', \mathcal{W}')\}.
\end{aligned}$$

Definition 41 (Program semantics)**.** Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be an $\mathcal{ALCOK}$-Golog program. The *transition system* $T_{\mathcal{P}} = (Q, \rightarrow, I)$ induced by $\mathcal{P}$ consists of the set of *states* Q as well as a set of *initial states* $I \subseteq Q$ with

$$Q := \mathfrak{I}_{\mathcal{K},\Sigma} \times \mathsf{subg}(\delta) \text{ and } I := \{\langle(\mathcal{I}, \mathcal{M}(\mathcal{K})), \delta\rangle \mid \mathcal{I} \in \mathcal{M}(\mathcal{K})\}$$

and a *labeled transition relation*

$$\rightarrow \subseteq Q \times \{\mathfrak{a} \mid \mathfrak{a} \in \mathsf{head}(\zeta) \text{ for some } \zeta \in \mathsf{subg}(\delta) \text{ or } \mathfrak{a} = \mathfrak{f}\} \times Q$$

such that

$$\langle(\mathcal{I}, \mathcal{W}), \rho\rangle \xrightarrow{\mathfrak{a}} \langle(\mathcal{I}', \mathcal{W}'), \rho'\rangle$$

iff one of the following conditions is satisfied:

1. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a} \in \Sigma_{\mathsf{g}}$ such that $\mathfrak{a} \neq \mathfrak{f}$, $(\mathcal{I}, \mathcal{W}) \Longrightarrow_{\mathfrak{a}} (\mathcal{I}', \mathcal{W}')$ and $\rho' \in \mathsf{tail}(\mathfrak{a}, \rho)$.
2. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a}$ is a test of the form $\psi?$, $(\mathcal{I}, \mathcal{W}) \models \psi$, $(\mathcal{I}', \mathcal{W}') = (\mathcal{I}, \mathcal{W})$ and $\rho' \in \mathsf{tail}(\psi?, \rho)$.
3. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a}$ is a pick of the form $\mathsf{pick}(\vec{x}) : \psi?$, there exists a variable mapping ν such that $(\mathcal{I}, \mathcal{W}) \models \psi^{\nu}$ and there exists $\zeta \in \mathsf{tail}(\mathfrak{a}, \rho)$ such that $\rho' = \zeta^{\nu}$, and $(\mathcal{I}', \mathcal{W}') = (\mathcal{I}, \mathcal{W})$.
4. No atomic program in $\mathsf{head}(\rho)$ is executable, $\mathfrak{a} = \mathfrak{f}$, $(\mathcal{I}, \mathcal{W}) \Longrightarrow_{\mathfrak{f}} (\mathcal{I}', \mathcal{W}')$ and $\rho' = \rho$.

▲

Due to the distinguished actions ϵ and $\mathfrak{f}$ a successor state is always guaranteed to exist. A *run* π of an $\mathcal{ALCOK}$-Golog program $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ is an infinite path in $T_{\mathcal{P}} = (Q, \rightarrow, I)$ of the form

$$\pi = \langle(\mathcal{I}_0, \mathcal{W}_0), \delta_0\rangle \xrightarrow{\mathfrak{a}_0} \langle(\mathcal{I}_1, \mathcal{W}_1), \delta_1\rangle \xrightarrow{\mathfrak{a}_1} \langle(\mathcal{I}_2, \mathcal{W}_2), \delta_2\rangle \xrightarrow{\mathfrak{a}_2} \cdots$$

starting in an initial state $\langle(\mathcal{I}_0, \mathcal{W}_0), \delta_0\rangle \in I$. The infinite sequence of epistemic interpretations occurring in the states along a run π is denoted by $\mathfrak{I}(\pi)$.

4.2 Specifying Temporal Properties of Programs

To specify temporal properties of a given program we use a logic we call $\mathcal{ALCOK}$-LTL. The syntax is the same as for propositional LTL, but in place of propositions we allow for $\mathcal{ALCOK}$-ABox assertions or $\mathcal{ALC}$-CIs. More precisely, $\mathcal{ALCOK}$-LTL formulas are built according to the following grammar:

$$\Phi ::= \varrho \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \mathsf{X}\Phi \mid \Phi_1 \,\mathsf{U}\, \Phi_2$$

where ϱ stands for an $\mathcal{ALCOK}$-ABox assertion or $\mathcal{ALC}$-CI. As usual, $\Diamond\Phi$ (*eventually*) and $\Box\Phi$ (*globally*) are used as abbreviations for $\top(a) \,\mathsf{U}\, \Phi$ and $\neg\Diamond\neg\Phi$, respectively.

The semantics of $\mathcal{ALCOK}$-LTL is based on the notion of an *$\mathcal{ALCOK}$-LTL structure*, which is an infinite sequence of epistemic interpretations $\mathfrak{I} = (\mathcal{I}_i, \mathcal{W}_i)_{i=0,1,2,\ldots}$. Let Φ be an $\mathcal{ALCOK}$-LTL formula, $\mathfrak{I}$ an $\mathcal{ALCOK}$-LTL structure, and $i \in \{0, 1, 2, \ldots\}$ a time point. Validity of Φ in $\mathfrak{I}$ at time i, denoted by $\mathfrak{I}, i \models \Phi$, is defined as follows:

$$\begin{array}{ll}
\mathfrak{I}, i \models \varrho & \text{iff } (\mathcal{I}_i, \mathcal{W}_i) \models \varrho, \\
\mathfrak{I}, i \models \neg\Phi & \text{iff } \mathfrak{I}, i \not\models \Phi, \\
\mathfrak{I}, i \models \Phi_1 \wedge \Phi_2 & \text{iff } \mathfrak{I}, i \models \Phi_1 \text{ and } \mathfrak{I}, i \models \Phi_2, \\
\mathfrak{I}, i \models \Phi_1 \vee \Phi_2 & \text{iff } \mathfrak{I}, i \models \Phi_1 \text{ or } \mathfrak{I}, i \models \Phi_2, \\
\mathfrak{I}, i \models \mathsf{X}\Phi & \text{iff } \mathfrak{I}, i+1 \models \Phi, \\
\mathfrak{I}, i \models \Phi_1 \,\mathsf{U}\, \Phi_2 & \text{iff } \exists k \geq i : \mathfrak{I}, k \models \Phi_2 \text{ and } \forall j, i \leq j < k : \mathfrak{I}, j \models \Phi_1
\end{array}$$

Now, we are ready to define the *verification problem.*

Definition 42 (verification problem)**.** Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be an $\mathcal{ALCOK}$-Golog program and Φ an $\mathcal{ALCOK}$-LTL formula. The formula Φ is *valid* in $\mathcal{P}$ iff for all runs π of $\mathcal{P}$ it holds that $\mathfrak{I}(\pi), 0 \models \Phi$. The formula Φ is *satisfiable* in $\mathcal{P}$ iff there exists a run π of $\mathcal{P}$ such that $\mathfrak{I}(\pi), 0 \models \Phi$. ▲

It holds that a formula Φ is valid in $\mathcal{P}$ iff $\neg\Phi$ is not satisfiable in $\mathcal{P}$.

Example 43. For a program in our example domain one might want to verify that the agent always knows whether *gear* is on or not, expressed by $\Box\mathbf{Kw}\,\mathit{On}(\mathit{gear})$. To ensure successful termination we can check whether $\Diamond\,\mathit{Term}(p)$ is valid. For the program expression given in Figure 1 and the initial KB shown in Figure 3 the following property is valid:

$$\exists \mathit{has\text{-}f.}(\mathit{CritFault} \sqcap \neg\mathbf{K}\mathit{Fault})(\mathit{gear}) \rightarrow \Diamond\mathbf{K}\exists \mathit{has\text{-}f.}(\mathit{CritFault} \sqcap \neg\mathbf{K}\mathit{Fault})(\mathit{gear})$$

saying that if *gear* has an unknown critical fault initially, then the agent will eventually recognize it. We can also verify executability and whether the TBox $\mathcal{T}$ is preserved along each run by checking validity of $\Box\big(\bigwedge_{\varrho\in\mathcal{T}} \varrho \wedge \neg\mathit{Fail}(p)\big)$. ▲

4.3 Undecidability of the Verification Problem

As expected the verification problem is in general undecidable. However we show that this holds for already very restricted subsets of our language. The main source of undecidability is the high degree of non-determinism introduced by the guarded pick operator that allows to quantify arguments ranging over the whole countably infinite domain Δ.

First, we reduce the entailment problem of boolean conjunctive queries (BCQs) w.r.t. $\mathcal{ALC}$-KBs to the validity problem of the verification problem. A *boolean conjunctive query* is of the form $\exists x_1 \cdots \exists x_m.\psi$ where $x_1, \ldots, x_m \in N_V$ are variables and ψ is a conjunction of primitive atoms such that $\mathsf{Var}(\psi) \subseteq \{x_1, \ldots, x_m\}$, i.e. all variables in ψ are existentially quantified.

A BCQ $q = \exists x_1 \cdots \exists x_m.\psi$ is satisfied in an interpretation $\mathcal{I}$ written as $\mathcal{I} \models q$ iff there exists a variable mapping ν such that $\mathcal{I} \models \psi^\nu$. We say that q is entailed by the $\mathcal{ALC}$-KB $\mathcal{K}$ iff q is satisfied in all models of $\mathcal{K}$. The entailment problem is undecidable even if q contains only four variables and $\mathcal{K}$ is restricted to the DL $\mathcal{AL}$ [Ros07]. $\mathcal{AL}$ is the sublogic of $\mathcal{ALC}$ where negation is only allowed in front of concept names and existential restrictions are restricted to be of the form $\exists r.\top$.

The reduction to the verification problem is straightforward. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ALC}$-KB and $q = \exists x_1 \cdots \exists x_m.\psi$ a BCQ. We define a program as follows: The initial KB is given by

$$\mathcal{K}' = (\mathcal{T}, \mathcal{A} \cup \{\neg\mathit{Term}(p), \neg\mathit{Fail}(p)\}).$$

and the program expression is given by

$$\delta = \mathsf{pick}(x_1, \ldots, x_m) : \psi?.[]$$

It easy to see that q is entailed by $\mathcal{K}$ iff $\Phi = \mathsf{X}\big(\mathsf{X}\,\mathit{Term}(p)\big)$ is valid in $\mathcal{P} = (\mathcal{K}', \Sigma = \{\epsilon, \mathfrak{f}\}, \delta)$.

Theorem 44. *The validity problem of the verification problem is undecidable even if the temporal property and the tests in the program are objective and no iteration is used.*

Next, we consider the class of knowledge-base programs. We show undecidability by a reduction of the halting problem of two-counter machines [Min67]. A two-counter machine M manipulates the non-negative integer values of two counters, denoted by c_0 and c_1 in the following. A machine M is given by a finite sequence of instructions of the form

$$\mathsf{M} = \mathsf{J}_0; \cdots ; \mathsf{J}_m.$$

Let $i, j \in \{0, \ldots, m\}$ and $n \in \{0, 1\}$. There are three kinds of instructions:

- $\mathsf{Inc}(n, i)$: Increment c_n by one and jump to instruction J_i.
- $\mathsf{Dec}(n, i, j)$: If $c_n = 0$ jump to J_i, else if $c_n > 0$ decrement c_n by one and jump to J_j.
- Halt: The machine stops.

A *configuration* of M is of the form (i, v_0, v_1) where $i \in \{0, \ldots, m\}$ is the index of the instruction to be executed next and $v_0, v_1 \in \mathbb{N}$ are the values of the two counters. M induces a transition relation on configurations, denoted by $\vdash_{\mathsf{M}}$, that is defined as explained above.

We assume that initially both counters are set to zero and the execution of M starts with instruction J_0. We say that M halts iff there exists a computation such that $(0, 0, 0) \vdash_{\mathsf{M}}{}^{*} (j, v_0, v_1)$ for some $v_0, v_1 \in \mathbb{N}$ and $\mathsf{J}_j = \mathsf{Halt}$. The problem of deciding whether a given two-counter machine halts or not is undecidable [Min67]. Simulating a two-counter machine with an $\mathcal{ALCOK}$-Golog program is straightforward.

Theorem 45. *The verification problem for knowledge-based programs is undecidable even if the initial TBox is empty, primitive actions have only unconditional effects, at most one argument, and provide no sensing result.*

Proof. Let $\mathsf{M} = \mathsf{J}_0; \cdots ; \mathsf{J}_m$ be a two-counter machine with the two counters c_0 and c_1. We construct a program $\mathcal{P}_{\mathsf{M}} = (\mathcal{K}_{\mathsf{M}} = (\mathcal{T}_{\mathsf{M}}, \mathcal{A}_{\mathsf{M}}), \Sigma_{\mathsf{M}}, \delta_{\mathsf{M}})$ and a $\mathcal{ALCOK}$-LTL formula Φ_{M} such that Φ_{M} is satisfiable in $\mathcal{P}_{\mathsf{M}}$ iff M reaches a halting configuration starting in the initial configuration $(0, 0, 0)$.

We use concept names $J_0, \ldots, J_m$, one for each instruction, and an individual $s \in \Delta$ with the intended meaning that s is a known instance of J_i iff J_i is the instruction that will be executed next. Furthermore, s is an instance of the concept name H iff M has reached a halting configuration. For the two counters c_0 and c_1 we introduce role names r_0 and r_1 and individuals a_0 and a_1. The value of the counter c_n with $n \in \{0, 1\}$ is represented in an epistemic interpretation as the number of known r_n-successors of the individual a_n. Thus, the initial configuration of M is represented in $\mathcal{K}_{\mathsf{M}}$ as follows:

$$\mathcal{K}_{\mathsf{M}} = (\mathcal{T}_{\mathsf{M}} = \emptyset, \mathcal{A}_{\mathsf{M}} = \{\forall r_0.\bot(a_0), \forall r_1.\bot(a_1), J_0(s)\}). \tag{4}$$

This means that it is known that a_0 and a_1 do not have any r_0-successors and r_1-successors, respectively. And $J_0(s)$ expresses that J_0 will be executed first.

Σ_{M} contains the following primitive actions: For each counter c_n with $n = 0, 1$ there is a unary increment and decrement action (without any sensing result).

$$\mathtt{inc}_n(x) = (\mathsf{eff} : \{r_n(a_n, x)\}) \text{ and } \mathtt{dec}_n(x) = (\mathsf{eff} : \{\neg r_n(a_n, x)\}). \tag{5}$$

Thus, an element x is added as an r_n-successor to a_n or is removed, respectively. For each instruction J_i with $i = 0, \ldots, m$ there is a jump action given by

$$\mathtt{jump}_i = (\mathsf{eff} : \{J_i(s)\} \cup \{\neg J_b(s) \mid b \neq i, 0 \leq b \leq m\}) \tag{6}$$

which determines the index of instruction that will be executed next. And there is an halting action:

$$\mathtt{halt} = (\mathsf{eff} : \{H(s)\}).$$

Next, we define for each instruction J_i with $i = 0, \ldots, m$ a program expression δ_i. If J_i is of the form $\mathsf{Inc}(n, j)$, then we define

$$\delta_i = \mathbf{K}J_i(s)?; \mathsf{pick}(x) : \mathbf{K}\neg r_n(a_n, x)?.\mathtt{inc}_n(x); \mathtt{jump}_j, \tag{7}$$

and if $\mathsf{J}_i = \mathsf{Dec}(n, j, j')$, then

$$\begin{aligned} \delta_i &= \mathbf{K}J_i(s)?; \big(\delta_{i,c_n=0} \mid \delta_{i,c_n>0}\big) \text{ with} \\ \delta_{i,c_n=0} &= \mathbf{K}\forall r_n.\bot(a_n)?; \mathtt{jump}_j \\ \delta_{i,c_n>0} &= \mathsf{pick}(x) : \mathbf{K}r_n(a_n, x)?.\mathtt{dec}_n(x); \mathtt{jump}_{j'} \end{aligned} \tag{8}$$

and if $\mathsf{J}_i = \mathsf{Halt}$, then $\delta_i = \mathbf{K}J_i(s)?; \mathtt{halt}$. Now, we can assemble the program as follows:

$$\delta_{\mathsf{M}} := \big(\delta_0 \mid \cdots \mid \delta_m\big)^* \tag{9}$$

Obviously, M halts iff $\Diamond\mathbf{K}H(s)$ is satisfiable in $\mathcal{P}_{\mathsf{M}}$. □

In the reduction above we have used role atoms with epistemic roles of the form $\mathbf{K}\neg r$ and concepts of the form $\forall r.\bot$. Next, we show that even if we disallow role names the verification problem stays undecidable.

Lemma 46. *The verification problem is undecidable even if role names are disallowed.*

Proof. We slightly modify the reduction given in Theorem 45. Again we use the concept names $J_0, \ldots, J_m$, H, the individual s and the actions $\mathtt{jump}_0, \ldots, \mathtt{jump}_m$ and $\mathtt{halt}$ as before.

For each counter c_n we use now three concept names C_n, $\widehat{C}_n$ and Z_n. The number of known instances of C_n represents the value of the counter c_n and if s is an instance of Z_n, then the value of c_n is zero.

The initial KB is given as follows:

$$\begin{aligned} \mathcal{K}_{\mathsf{M}} = (\mathcal{T}_{\mathsf{M}} &= \{C_0 \sqsubseteq \bot, \top \sqsubseteq \widehat{C}_0, C_1 \sqsubseteq \bot, \top \sqsubseteq \widehat{C}_1\}, \\ \mathcal{A}_{\mathsf{M}} &= \{J_0(s), Z_0(s), Z_1(s)\}). \end{aligned} \tag{10}$$

The increment actions are now defined by

$$\mathtt{inc}_n(x) = (\mathsf{eff} : \{\neg\widehat{C}_n(x), C_n(x), \neg Z_n(s)\}). \tag{11}$$

Thus, we shift an element x from $\widehat{C}_n$ to C_n. After incrementing c_n the value cannot be equal to zero therefore $Z_n(s)$ is set to false. There are two different kinds of decrement actions:

$$\begin{aligned} \mathtt{dec}_n(x) &= (\mathsf{eff} : \{\neg C_n(x), \widehat{C}_n(x)\}) \text{ and} \\ \mathtt{decz}_n(x) &= (\mathsf{eff} : \{\neg C_n(x), \widehat{C}_n(x), Z_n(s)\}). \end{aligned} \tag{12}$$

To decrement c_n an element x is shifted from C_n to $\widehat{C}_n$. In case $c_n = 0$ after decrementing c_n the second version of the action is chosen.

For an instruction $\mathsf{J}_i = \mathsf{Inc}(c_n, j)$ we use the program expression

$$\delta_i := \mathbf{K}J_i(s)?; \mathsf{pick}(x) : \mathbf{K}\widehat{C}(x)?.\mathtt{inc}_n(x); \mathtt{jump}_j. \tag{13}$$

For $\mathsf{J}_i = \mathsf{Dec}(n, j, j')$ we have

$$\begin{aligned} &\delta_i := \mathbf{K}J_i(s)?; \big(\delta_{i,c_n=0} \mid \delta_{i,c_n>0}\big) \text{ with} \\ &\delta_{i,c_n=0} := \mathbf{K}Z_n(s)?; \mathtt{jump}_j \\ &\delta_{i,c_n>0} := \mathsf{pick}(x) : \mathbf{K}C_n(x)?.\big(\mathtt{dec}_n(x) \mid \mathtt{decz}_n(x)\big); \mathtt{jump}_{j'} \end{aligned} \tag{14}$$

The program expression for a halting instruction and δ_{M} are defined as before. The property Φ_{M} is defined as follows

$$\Box(\mathbf{K}Z_0(s) \rightarrow C_0 \sqsubseteq \bot) \wedge \Box(\mathbf{K}Z_1(s) \rightarrow C_1 \sqsubseteq \bot) \wedge \Diamond\mathbf{K}H(s).$$

Obviously, M halts iff Φ_{M} is satisfiable in $\mathcal{P}_{\mathsf{M}}$. □

In this section we have identified the non-deterministic guarded pick construct as the main source of undecidability. One possible restriction to achieve decidability could be to consider weaker variants of the verification problem. For instance we could introduce a fixed finite horizon on the transition system and consider only finite prefixes of runs up to a given length k instead of examining the complete transition system. However as we have seen in the reduction of the BCQ entailment problem such a boundedness restriction alone does not suffice for achieving decidability. The reason is that the initial TBox and the test used as guard in the guarded pick expression can be used to enforce complex structures of models such as domino tilings used in Rosati's undecidability proof [Ros07] of BCQ entailment. The changes caused by actions are not important here. In the cases considered in Theorem 45 and Lemma 46 the source of undecidability is of a quite different nature. Here the problem is that the pick inside the loop allows us to inject an unbounded number of new objects as known instances of a concept such that the knowledge base of the agent may grow indefinitely.

4.4 Decidable Verification of Restricted Programs

In order to retain decidability of the verification problem for $\mathcal{ALCOK}$-Golog programs we restrict the *syntax of the guards* allowed in the pick-operators. With this restriction we achieve that only known individuals, i.e. those individuals that have a name given in the program, can be chosen for instantiation.

Definition 47 (restricted programs)**.** Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be an $\mathcal{ALCOK}$-Golog program. $\mathcal{P}$ is called *restricted* if all guarded picks occurring in δ are of the form

$$\mathsf{pick}(x_1, \ldots, x_n) : (\psi_1 \vee \cdots \vee \psi_m) \wedge \psi'?$$

with $m \geq 1$ and for each ψ_i with $1 \leq i \leq m$ it holds that $\{x_1, \ldots, x_n\} \subseteq \mathsf{Var}(\psi_i)$ and ψ_i is a conjunction of concept and role atoms of the form $\mathbf{K}C(z)$ and $\mathbf{K}r(z, z')$, respectively, where C is an $\mathcal{ALC}$-concept with $\mathcal{K} \not\models \top \sqsubseteq C$ and $r \in N_R$. There are no restrictions on the formula ψ'. ▲

Note that the example program in Figure 1 is restricted. The restricted atoms in the guard of the pick-operator can also be viewed as objective instance queries posed to the current KB of the agent. The agent then chooses a binding for the variables among the retrieved answers.

We argue that the restrictions imposed on the first conjunct $(\psi_1 \vee \cdots \vee \psi_m)$ of the guards are necessary to achieve decidability in the sense that dropping one of these restrictions leads to undecidability again:

- Allowing in addition also atoms of the form $\mathsf{K}\neg r(x,y)$ or $\neg\mathsf{K}r(x,y)$ causes undecidability: The program in the proof of Theorem 45 satisfies all the restriction except for the guard of the increment action (see (7)) where the variable x only appears in the atom $\mathsf{K}\neg r_n(a_n, x)$. The same reduction also works if we replace $\mathsf{K}\neg r_n(a_n, x)$ by $\neg\mathsf{K}r_n(a_n, x)$. Thus, allowing in addition also negated subjective atoms leads to undecidability.
- The condition $\mathcal{K} \not\models \top \sqsubseteq C$ for concept atoms $\mathsf{K}C(x)$ is also necessary. The program in the proof of Lemma 46 satisfies all the restrictions except for the guard $\mathsf{K}\widehat{C}_n(x)$? of the increment action (see (13)) where it holds that $\mathcal{K}_{\mathsf{M}} \models \top \sqsubseteq \widehat{C}_n$.
- If we allow nominals or nesting of the K-constructor in concept atoms, then any knowledge-based program can be equivalently modified into a restricted one as we have seen in Example 26. Similarly, dropping the condition $\{x_1, \ldots, x_n\} \subseteq \mathsf{Var}(\psi_i)$ would allow us to undermine all the other restriction.

As shown in Lemma 22 concepts like $\mathsf{K}C$ where C is nominal- and K-free and not equivalent to $\top$ and roles of the form $\mathsf{K}r$ are interpreted as finite subsets of Ind and $\mathsf{Ind} \times \mathsf{Ind}$, respectively, under an epistemic interpretation that can be reached from the initial epistemic model by performing a sequence of ground actions where Ind are the individuals mentioned in the actions or in the initial KB. As a consequence we get that the agent is allowed to only choose objects from the set of known individuals as formulated in the next lemma. From now on Ind denotes the set of all individuals occurring in $\mathcal{P}$.

Lemma 48. *Let $\langle(\mathcal{I}, \mathcal{W}), \rho\rangle \in Q$ be a state reachable from an initial state in the transition system $T_{\mathcal{P}} = (Q, \rightarrow, I)$ of a restricted $\mathcal{ALCOK}$-Golog program $\mathcal{P}$. It holds that ρ contains only individuals from* Ind.

Proof. We show this claim for all states in Q reachable from an initial state by induction on the length n of a shortest initial fragment of a run leading to the state. Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ and $\langle(\mathcal{I}, \mathcal{W}), \rho\rangle \in Q$. First assume $n = 0$. Consequently, $\langle(\mathcal{I}, \mathcal{W}), \rho\rangle$ is an initial state with $\rho = \delta$. By definition of Ind δ contains only individuals from Ind. Consider $n > 0$. There exists the initial fragment of a run of the form:

$$\langle(\mathcal{I}_1, \mathcal{W}_1), \delta_1\rangle \overset{\mathfrak{a}_1}{\rightarrow} \langle(\mathcal{I}_2, \mathcal{W}_2), \delta_2\rangle \overset{\mathfrak{a}_2}{\rightarrow} \cdots \overset{\mathfrak{a}_{n-1}}{\rightarrow} \langle(\mathcal{I}_n, \mathcal{W}_n), \delta_n\rangle$$

with $\langle(\mathcal{I}, \mathcal{W}), \rho\rangle = \langle(\mathcal{I}_n, \mathcal{W}_n), \delta_n\rangle$. By induction the program expressions $\delta_1, \ldots, \delta_{n-1}$ contain only individuals from Ind and are closed. Assume that $\mathfrak{a}_{n-1}$ is a primitive action, test or the failing action. It follows that $\delta_n \in \mathsf{tail}(\mathfrak{a}_{n-1}, \delta_{n-1})$. Since δ_{n-1} contains only individuals from Ind also the tails of δ_{n-1} including δ_n only contain individuals from Ind. For the remaining case assume that $\mathfrak{a}_{n-1}$ is a guarded pick of the form:

$$\mathsf{pick}(x_1, \ldots, x_m) : (\psi_1 \vee \cdots \vee \psi_\ell) \wedge \psi'?.$$

There exists a variable mapping ν and a $k \in \{1, \ldots, \ell\}$ such that $(\mathcal{I}_{n-1}, \mathcal{W}_{n-1}) \mid\models {\psi_k}^{\nu}$. Since all program expression $\delta_1, \ldots, \delta_{n-1}$ contain only individuals from Ind also all the heads of them contain only individuals from Ind. Therefore, there exists a sequence of ground actions σ with individuals only from Ind such that $(\mathcal{I}_1, \mathcal{W}_1) \Longrightarrow_{\sigma} (\mathcal{I}_{n-1}, \mathcal{W}_{n-1})$. We have that $\{x_1, \ldots, x_m\} = \mathsf{Var}(\psi_k)$ and ψ_k is a conjunction of atoms of the form $\mathsf{K}C(x)$ or $\mathsf{K}r(x,y)$ where C is an $\mathcal{ALC}$-concept with $\mathcal{K} \not\models \top \sqsubseteq C$ by definition of the restricted guards. Since $(\mathcal{I}_{n-1}, \mathcal{W}_{n-1}) \mid\models {\psi_k}^{\nu}$, for each variable $x_j \in \{x_1, \ldots, x_m\}$ it holds that there exists a concept $\mathsf{K}C$ and $\nu(x_j) \in \mathsf{K}C^{\mathcal{W}_{n-1}}$

or there exists a role $\mathsf{K}r$ and $x_{j'} \in \{x_1, \ldots, x_m\}$ such that $(\nu(x_j), \nu(x_{j'})) \in \mathsf{K}r^{\mathcal{W}_{n-1}}$. From Lemma 22.1 and 22.2 it follows that $\nu(x_j) \in \mathsf{Ind}$. Finally, there exists

$$\zeta \in \mathsf{tail}(\mathsf{pick}(x_1, \ldots, x_m) : (\psi_1 \vee \cdots \vee \psi_\ell) \wedge \psi'?, \delta_{n-1})$$

such that $\delta_n = \zeta^\nu$. Since δ_{n-1} is closed and contains only individuals from Ind, it follows that ζ from the tail also contains only individuals from Ind and the free variables in ζ are among the set $\{x_1, \ldots, x_m\}$. As shown above all variables from $\{x_1, \ldots, x_m\}$ are mapped to some individual in Ind by ν. Therefore $\delta = \zeta^\nu$ only contains individuals from Ind. □

As a consequence we obtain that the (reachable fragment of the) transition system of a restricted program has a finite branching degree bounded by the number of individuals in Ind. Furthermore, from Lemma 39 and Lemma 48 it follows that the number of reachable program expressions in the transition system is polynomially bounded in the size of δ and Ind. However the transition system $T_{\mathcal{P}} = (Q, \rightarrow, I)$ still has infinitely many initial states. To reduce the verification problem to model checking we define an equivalence relation on epistemic interpretations with a finite number of equivalence classes that we call types. We start with identifying a finite set of relevant (possibly epistemic) axioms, which we call *context* of a program $\mathcal{P}$. For this we consider all possible groundings of atoms that appear in the program. Given an atom φ, a variable mapping ν is called *grounding for* φ if for all $x \in \mathsf{Var}(\varphi)$ it holds that $\nu(x) \in \mathsf{Ind}$. In addition to ABox assertions and CIs we also consider *negated CIs* of the form $\neg(C \sqsubseteq D)$. Satisfaction of a negated CI in an interpretation is defined in the obvious way.

Definition 49 (context)**.** Let $\mathcal{P} = (\mathcal{K} = (\mathcal{T}, \mathcal{A}), \Sigma, \delta)$. The *context* of $\mathcal{P}$, denoted by $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$, is defined as the least set satisfying the following conditions:

- $\mathcal{T} \subseteq \mathcal{C}_{\mathsf{K}}(\mathcal{P})$ and $\mathcal{A} \subseteq \mathcal{C}_{\mathsf{K}}(\mathcal{P})$;
- If φ is an atom occurring in $\mathcal{P}$ and ν is a grounding for φ, then $\varphi^\nu \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$.
- If $r \in N_R$ is a role name occurring in $\mathcal{P}$ and $a, b \in \mathsf{Ind}$, then $r(a, b) \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$.
- If $\varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$, then also $\neg\varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$.

▲

Now we extend the context with a finite set of non-epistemic axioms by considering all relevant rewritings of subjective sub-concepts and roles. The objective nominal concepts that replaces an epistemic sub-concept is determined by an instance function. We define a rewriting operator as a small modification of $[\![\cdot, \cdot]\!]$ (see Figure 4). Let C be an $\mathcal{ALCOK}$-concept and κ an instance function. The $\mathcal{ALCO}$-concept $[C, \kappa]$ is defined inductively on the structure of C exactly in the same way as $[\![C, \kappa]\!]$ except for the case $C = \mathsf{K}D$ where instead of $[\![\mathsf{K}D, \kappa]\!] := \bigsqcup \kappa(\mathsf{K}[\![D, \kappa]\!])$ we define $[\mathsf{K}D, \kappa] := \bigsqcup \kappa(\mathsf{K}D)$. We show that both operators yield the same result.

Lemma 50. *Let C be an $\mathcal{ALCOK}$-concept and $\langle(\mathcal{J}, \mathcal{W}), \rho\rangle \in Q$ a reachable state in $T_{\mathcal{P}}$. It holds that $[\![C, \kappa_{\mathcal{W}}]\!] = [C, \kappa_{\mathcal{W}}]$.*

Proof. The claim is shown by induction on the structure of C. Consider the case where C is of the form $\mathsf{K}D$. By definition of $T_{\mathcal{P}}$ there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and a sequence of primitive ground actions σ such that $(\mathcal{I}, \mathcal{M}(\mathcal{K})) \Longrightarrow_\sigma (\mathcal{J}, \mathcal{W})$. Therefore, using the definition of the instance function and Lemma 28 we obtain

$$\begin{aligned}
[\![\mathsf{K}D, \kappa_{\mathcal{W}}]\!] :&= \bigsqcup \kappa_{\mathcal{W}}(\mathsf{K}[\![D, \kappa_{\mathcal{W}}]\!]) \\
&= \bigsqcup \{\{a\} \mid a \in \bigcap_{\mathcal{J}\in\mathcal{W}} [\![D, \kappa_{\mathcal{W}}]\!]^{\mathcal{J},\mathcal{W}}, a \in \mathsf{Ind}\} \cup \\
&\qquad \{\neg N \mid \exists c : c \in \Delta \setminus \mathsf{Ind}, c \in \bigcap_{\mathcal{J}\in\mathcal{W}} [\![D, \kappa_{\mathcal{W}}]\!]^{\mathcal{J},\mathcal{W}}\} \\
&= \bigsqcup \{\{a\} \mid a \in \bigcap_{\mathcal{J}\in\mathcal{W}} [\![D, \kappa_{\mathcal{W}}]\!]^{\mathcal{J}}, a \in \mathsf{Ind}\} \cup \\
&\qquad \{\neg N \mid \exists c : c \in \Delta \setminus \mathsf{Ind}, c \in \bigcap_{\mathcal{J}\in\mathcal{W}} [\![D, \kappa_{\mathcal{W}}]\!]^{\mathcal{J}}\} \\
&= \bigsqcup \{\{a\} \mid a \in \bigcap_{\mathcal{J}\in\mathcal{W}} D^{\mathcal{J},\mathcal{W}}, a \in \mathsf{Ind}\} \cup \\
&\qquad \{\neg N \mid \exists c : c \in \Delta \setminus \mathsf{Ind}, c \in \bigcap_{\mathcal{J}\in\mathcal{W}} D^{\mathcal{J},\mathcal{W}}\} \\
&= \bigsqcup \kappa_{\mathcal{W}}(\mathsf{K}D) \\
&= [\mathsf{K}D, \kappa_{\mathcal{W}}].
\end{aligned}$$

The remaining cases are trivial. □

Using the rewriting operator we close up the context under all possible rewritings of epistemic subconcepts occurring in the context.

Definition 51. Let $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ be the context of a program $\mathcal{P}$ and $c_u \in \Delta \setminus \mathsf{Ind}$ an anonymous individual. Furthermore, let $\mathfrak{F}$ be the set of all instance functions. The *knowledge closure* of $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$, denoted by $\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$, is defined as follows:

$$\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u) := \mathcal{C}_{\mathsf{K}}(\mathcal{P}) \;\cup \bigcup_{\substack{C(a)\in\mathcal{C}_{\mathsf{K}}(\mathcal{P})\\ \kappa\in\mathfrak{F}}} \begin{array}{l}\{[C,\kappa](a), \neg[C,\kappa](a)\} \cup \\ \{[D,\kappa](b), \neg[D,\kappa](b) \mid \mathsf{K}D \in \mathsf{sub}(C), b \in \mathsf{Ind} \cup \{c_u\}\}.\end{array}$$

The subset of $\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$ containing all K-free axioms contained in $\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$ is denoted by $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$. ▲

Note, that the distinguished individual $c_u \in \Delta \setminus \mathsf{Ind}$ will be used to indicate whether there exists an anonymous element in $\mathsf{K}D$ or not. To handle epistemic roles we have already included all assertions of the form $r(a, b)$ and $\neg r(a, b)$ with $a, b \in \mathsf{Ind}$ (see third item Def. 49) in the context.

Lemma 52. *$\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$ is at most exponentially large in the size of $\mathcal{P}$.*

Proof. For a fixed instance function $\kappa \in \mathfrak{F}$ there are polynomially many new K-free assertions that are added to the context $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. One for each concept assertion $C(a)$ in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ and $|\mathsf{Ind}| + 1$ many for each subconcept of the form $\mathsf{K}D$ occurring in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. Each new assertion is of polynomial size in the size of C and D, respectively, and Ind.

Consider the following sets

$$\begin{aligned}
S_C &:= \{\mathsf{K}D \mid C(a) \in \mathcal{C}_{\mathsf{K}}(\mathcal{P}), \mathsf{K}D \in \mathsf{sub}(C)\} \text{ and} \\
S_R &:= \{(a, r) \mid a \in \mathsf{Ind}, r \in N_R \text{ and } r \text{ occurs in } \mathcal{C}_{\mathsf{K}}(\mathcal{P})\}.
\end{aligned}$$

Let $\kappa \in \mathfrak{F}$ be an instance function. The instance function $\kappa_{\restriction S_C \cup S_R}$ is defined as κ but restricted to the domain $S_C \cup S_R$. $\mathfrak{F}_{\restriction S_C \cup S_R}$ denotes the set of all restricted instance functions. It is easy to see that we obtain exactly the same set if we replace $\mathfrak{F}$ by $\mathfrak{F}_{\restriction S_C \cup S_R}$ in the definition of $\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$. There are exponentially many different instance functions in $\mathfrak{F}_{\restriction S_C \cup S_R}$ in the size of $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ and Ind. Each function in $\kappa \in \mathfrak{F}_{\restriction S_C \cup S_R}$ is a total function of the form:

$$\kappa : S_C \cup S_R \to 2^{\{\{a\} \mid a \in \mathsf{Ind}\} \cup \{\neg N\}} \cup 2^{\{\{a\} \mid a \in \mathsf{Ind}\}}.$$

Consequently,

$$\begin{aligned}
&|\mathfrak{F}_{\restriction S_C \cup S_R}| = n^m \text{ with} \\
&n = |2^{\{\{a\} \mid a \in \mathsf{Ind}\} \cup \{\neg N\}} \cup 2^{\{\{a\} \mid a \in \mathsf{Ind}\}}| \text{ and} \\
&m = |S_C \cup S_R|.
\end{aligned}$$

Therefore, the size of the knowledge closure $\widehat{\mathcal{C}}_{\mathsf{K}}(\mathcal{P}, c_u)$ is exponential in the size of $\mathcal{P}$ and Ind. □

The two sets of axioms $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ and $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ are used to partition epistemic interpretations and interpretations into finitely many equivalence classes which we call static types.

Definition 53 (static types)**.** Let $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ be the context of a program $\mathcal{P}$ and $(\mathcal{I}, \mathcal{W})$ an epistemic interpretation with $\mathcal{I} \in \mathcal{W}$ and $\mathcal{J} = (\Delta, \cdot^{\mathcal{J}})$ an interpretation. The *epistemic static type* of $(\mathcal{I}, \mathcal{W})$ w.r.t. $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$, denoted by $\mathsf{s\text{-}type}_{\mathsf{K}}(\mathcal{I}, \mathcal{W})$, is given by:

$$\mathsf{s\text{-}type}_{\mathsf{K}}(\mathcal{I}, \mathcal{W}) := \{\varphi \in \mathcal{C}_{\mathsf{K}}(\mathcal{P}) \mid (\mathcal{I}, \mathcal{W}) \mid\!\models \varphi\}.$$

The *non-epistemic static type* of a single interpretation $\mathcal{J}$ w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ is given by:

$$\mathsf{s\text{-}type}(\mathcal{I}) := \{\varphi \in \widehat{\mathcal{C}}(\mathcal{P}, c_u) \mid \mathcal{I} \models \varphi\}.$$

▲

From the definition it follows that for an epistemic static type $\mathfrak{s}_{\mathsf{K}} = \mathsf{s\text{-}type}_{\mathsf{K}}(\mathcal{I}, \mathcal{W})$ it holds that $\varrho \in \mathfrak{s}_{\mathsf{K}}$ iff $\neg\varrho \notin \mathfrak{s}_{\mathsf{K}}$ for all $\varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$. And likewisefor a static type $\mathfrak{s} = \mathsf{s\text{-}type}(\mathcal{I})$ it holds that $\varrho \in \mathfrak{s}$ iff $\neg\varrho \notin \mathfrak{s}$ for all $\varrho \in \widehat{\mathcal{C}}(\mathcal{P}, c_u)$. Note that $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ and $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ are closed under negation and we assume ($\neg\neg\varrho := \varrho$).

In the following we show that the *epistemic* static type of $(\mathcal{I}, \mathcal{W})$ is uniquely determined by the *non-epistemic static type* of $\mathcal{I}$ and by the *non-epistemic static types* of the interpretations in $\mathcal{W}$. First, we define an instance function based on a set of non-epistemic static types $\mathfrak{S}$.

$$\begin{aligned}
\kappa_{\mathfrak{S}}(\mathsf{K}D) &:= \{\{a\} \mid D(a) \in \textstyle\bigcap \mathfrak{S}\} \cup \{\neg N \mid D(c_u) \in \textstyle\bigcap \mathfrak{S}\} \\
\kappa_{\mathfrak{S}}(a, r) &:= \{\{b\} \mid r(a, b) \in \textstyle\bigcap \mathfrak{S}\} \\
\kappa_{\mathfrak{S}}(a, \neg r) &:= \{\{b\} \mid \neg r(a, b) \in \textstyle\bigcap \mathfrak{S}\}.
\end{aligned}$$

Next we show that the instance function $\kappa_{\mathcal{W}}$ can be equivalently replaced by $\kappa_{\mathfrak{S}}$ with

$$\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\}.$$

Lemma 54. *Let $\langle (\mathcal{I}, \mathcal{W}), \rho \rangle$ be a reachable state in $T_{\mathcal{P}}$ and*

$$\mathfrak{s} = \mathsf{s\text{-}type}(\mathcal{I}) \textit{ and } \mathfrak{S} = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\}$$

the corresponding static types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$.

1. *Let $a \in \mathsf{Ind}$ and $r \in N_R$. It holds that $\kappa_{\mathcal{W}}(a,r) = \kappa_{\mathfrak{S}}(a,r)$.*
2. *Let $D \in \mathsf{sub}(C)$ for some $C(a) \in \mathcal{C}_{\mathbf{K}}(\mathcal{P})$. It holds that $\llbracket D, \kappa_{\mathcal{W}} \rrbracket = \llbracket D, \kappa_{\mathfrak{S}} \rrbracket$.*
3. *Let $C(a) \in \mathcal{C}_{\mathbf{K}}(\mathcal{P})$. It holds that*

$$(\mathcal{I}, \mathcal{W}) \mid\!\models C(a) \textit{ iff } C(a) \in \mathsf{s\text{-}type}_{\mathbf{K}}(\mathcal{I}, \mathcal{W}) \textit{ iff } \llbracket C, \kappa_{\mathfrak{S}} \rrbracket(a) \in \mathfrak{s}.$$

Proof.

1. It holds that $\{b\} \in \kappa_{\mathcal{W}}(a,r)$

   iff $(a,b) \in \bigcap_{\mathcal{J} \in \mathcal{W}} r^{\mathcal{J}}$
   iff $\mathcal{J} \models r(a,b)$ for all $\mathcal{J} \in \mathcal{W}$
   iff $r(a,b) \in \mathsf{s\text{-}type}(\mathcal{J})$ for all $\mathcal{J} \in \mathcal{W}$ (since $r(a,b) \in \mathcal{C}_{\mathbf{K}}(\mathcal{P})$)
   iff $r(a,b) \in \bigcap_{\mathcal{J} \in \mathcal{W}} \mathsf{s\text{-}type}(\mathcal{J})$
   iff $\{b\} \in \kappa_{\mathfrak{S}}(a,r)$.

2. Consider the following set of concepts:

$$\mathsf{sub}(\mathcal{P}) := \{D \mid D \in \mathsf{sub}(C), C(a) \in \mathcal{C}_{\mathbf{K}}(\mathcal{P})\}.$$

   By induction we prove that for all concept $C \in \mathsf{sub}(\mathcal{P})$ it holds that $\llbracket C, \kappa_{\mathcal{W}} \rrbracket = \llbracket C, \kappa_{\mathfrak{S}} \rrbracket$. For concepts of the form A, $\{a\}$, $\top$ or $\bot$ in $\mathsf{sub}(\mathcal{P})$ the claim trivially holds.

   $C = \neg D$: Obviously, $\neg D \in \mathsf{sub}(\mathcal{P})$ implies $D \in \mathsf{sub}(\mathcal{P})$. The induction hypothesis yields $\llbracket D, \kappa_{\mathcal{W}} \rrbracket = \llbracket D, \kappa_{\mathfrak{S}} \rrbracket$. Using the definition of $\llbracket \cdot, \cdot \rrbracket$ we obtain

$$\llbracket \neg D, \kappa_{\mathcal{W}} \rrbracket = \neg \llbracket D, \kappa_{\mathcal{W}} \rrbracket = \neg \llbracket D, \kappa_{\mathfrak{S}} \rrbracket = \llbracket \neg D, \kappa_{\mathfrak{S}} \rrbracket.$$

   $C = D_1 \sqcap D_2, \exists r.D$: The proof is analogous to the previous case using the induction hypothesis and the definition of $\llbracket \cdot, \cdot \rrbracket$.

   $C = \exists \mathbf{K} r.D$: By claim 1 it holds that $\kappa_{\mathcal{W}}(a,r) = \kappa_{\mathfrak{S}}(a,r)$ for all $a \in \mathsf{Ind}$. It holds that $D \in \mathsf{sub}(\mathcal{P})$. Therefore, the induction hypothesis yields $\llbracket D, \kappa_{\mathcal{W}} \rrbracket = \llbracket D, \kappa_{\mathfrak{S}} \rrbracket$. It follows that $\llbracket \exists \mathbf{K} r.D, \kappa_{\mathcal{W}} \rrbracket = \llbracket \exists \mathbf{K} r.D, \kappa_{\mathfrak{S}} \rrbracket$.

   $C = \mathbf{K} D$: Since $\mathbf{K} D \in \mathsf{sub}(\mathcal{P})$, we have $D \in \mathsf{sub}(\mathcal{P})$. Using Lemma 50 and the induction hypothesis we obtain:

$$[D, \kappa_{\mathcal{W}}] \overset{\text{L. 50}}{=} \llbracket D, \kappa_{\mathcal{W}} \rrbracket \overset{\text{i. h.}}{=} \llbracket D, \kappa_{\mathfrak{S}} \rrbracket. \tag{15}$$

   Furthermore, by definition of $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ it holds that $\mathbf{K} D \in \mathsf{sub}(\mathcal{P})$ implies

$$[D, \kappa_{\mathcal{W}}](b) \in \widehat{\mathcal{C}}(\mathcal{P}, c_u) \text{ for all } b \in \mathsf{Ind} \cup \{c_u\}. \tag{16}$$

   By definition it holds that

$$\llbracket \mathbf{K} D, \kappa_{\mathcal{W}} \rrbracket = \bigsqcup \kappa_{\mathcal{W}}(\mathbf{K} \llbracket D, \kappa_{\mathcal{W}} \rrbracket) \text{ and } \llbracket \mathbf{K} D, \kappa_{\mathfrak{S}} \rrbracket = \bigsqcup \kappa_{\mathfrak{S}}(\mathbf{K} \llbracket D, \kappa_{\mathfrak{S}} \rrbracket).$$

   We show $\kappa_{\mathcal{W}}(\mathbf{K} \llbracket D, \kappa_{\mathcal{W}} \rrbracket) = \kappa_{\mathfrak{S}}(\mathbf{K} \llbracket D, \kappa_{\mathfrak{S}} \rrbracket)$ as follows.
   It holds that $\{a\} \in \kappa_{\mathcal{W}}(\mathbf{K} \llbracket D, \kappa_{\mathcal{W}} \rrbracket)$ for an $a \in \mathsf{Ind}$

iff $a \in \bigcap\limits_{\mathcal{J}\in\mathcal{W}} \llbracket D, \kappa_{\mathcal{W}} \rrbracket^{\mathcal{J},\mathcal{W}}$
iff $a \in \bigcap\limits_{\mathcal{J}\in\mathcal{W}} \llbracket D, \kappa_{\mathcal{W}} \rrbracket^{\mathcal{J}}$
iff $a \in \bigcap\limits_{\mathcal{J}\in\mathcal{W}} [D, \kappa_{\mathcal{W}}]^{\mathcal{J}}$ (with (15))
iff $\mathcal{J} \models [D, \kappa_{\mathcal{W}}](a)$ for all $\mathcal{J} \in \mathcal{W}$
iff $[D, \kappa_{\mathcal{W}}](a) \in \mathsf{s\text{-}type}(\mathcal{J})$ for all $\mathcal{J} \in \mathcal{W}$ (with (16))
iff $\llbracket D, \kappa_{\mathfrak{S}} \rrbracket(a) \in \mathsf{s\text{-}type}(\mathcal{J})$ for all $\mathcal{J} \in \mathcal{W}$ (with (15))
iff $\{a\} \in \kappa_{\mathfrak{S}}(\mathbf{K}\llbracket D, \kappa_{\mathfrak{S}} \rrbracket)$.

Similarly, it holds that $\neg N \in \kappa_{\mathcal{W}}(\mathbf{K}\llbracket D, \kappa_{\mathcal{W}} \rrbracket)$
iff there exists $c \in \Delta \setminus \mathsf{Ind}$ such that $c \in (\mathbf{K}\llbracket D, \kappa_{\mathcal{W}} \rrbracket)^{\mathcal{W}}$
iff $\Delta \setminus \mathsf{Ind} \subseteq (\mathbf{K}\llbracket D, \kappa_{\mathcal{W}} \rrbracket)^{\mathcal{W}}$ (by Lemma 18)
iff there exists a $c_u \in \Delta \setminus \mathsf{Ind}$ such that $\mathcal{J} \models \llbracket D, \kappa_{\mathcal{W}} \rrbracket(c_u)$ for all $\mathcal{J} \in \mathcal{W}$
iff there exists a $c_u \in \Delta \setminus \mathsf{Ind}$ such that $\mathcal{J} \models [D, \kappa_{\mathcal{W}}](c_u)$ for all $\mathcal{J} \in \mathcal{W}$ (with (15))
iff $[D, \kappa_{\mathcal{W}}](c_u) \in \mathsf{s\text{-}type}(\mathcal{J})$ for all $\mathcal{J} \in \mathcal{W}$ (with (16))
iff $\llbracket D, \kappa_{\mathfrak{S}} \rrbracket(c_u) \in \mathsf{s\text{-}type}(\mathcal{J})$ for all $\mathcal{J} \in \mathcal{W}$ (with (15))
iff $\neg N \in \kappa_{\mathfrak{S}}(\mathbf{K}\llbracket D, \kappa_{\mathfrak{S}} \rrbracket)$.

3. It holds that $(\mathcal{I}, \mathcal{W}) \mid\!\models C(a)$

iff $C(a) \in \mathsf{s\text{-}type}_{\mathbf{K}}(\mathcal{I}, \mathcal{W})$ (by definition of static type)
iff $a \in C^{\mathcal{I},\mathcal{W}}$
iff $a \in \llbracket C, \kappa_{\mathcal{W}} \rrbracket^{\mathcal{I}}$ (by Lemma 28)
iff $\llbracket C, \kappa_{\mathcal{W}} \rrbracket(a) \in \mathfrak{s}$ (by Lemma 50: $\llbracket C, \kappa_{\mathcal{W}} \rrbracket = [C, \kappa_{\mathcal{W}}]$ and by definition of the knowledge closure it holds that $[C, \kappa_{\mathcal{W}}](a) \in \widehat{\mathcal{C}}(\mathcal{P}, c_u)$)
iff $\llbracket C, \kappa_{\mathfrak{S}} \rrbracket(a) \in \mathfrak{s}$ (by claim 2).

□

Similarly, for role assertions of the form $\mathbf{K}r(a, b) \in \mathcal{C}_{\mathbf{K}}(\mathcal{P})$ it holds that

$$\mathbf{K}r(a, b) \in \mathsf{s\text{-}type}_{\mathbf{K}}(\mathcal{I}, \mathcal{W}) \text{ iff } \{b\} \in \kappa_{\mathfrak{S}}(a, r)$$

with $\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\}$ and

$$\mathbf{K}\neg r(a, b) \in \mathsf{s\text{-}type}_{\mathbf{K}}(\mathcal{I}, \mathcal{W}) \text{ iff } \{b\} \in \kappa_{\mathfrak{S}}(a, \neg r).$$

For the corresponding abstraction of an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ given as a pair $(\mathfrak{s}, \mathfrak{S})$ with $\mathfrak{s} = \mathsf{s\text{-}type}(\mathcal{I})$ and $\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\}$ we define satisfaction of an axiom from $\mathcal{C}_{\mathbf{K}}(\mathcal{P})$ in $(\mathfrak{s}, \mathfrak{S})$ as follows:

$$\begin{aligned}
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models C(a) \text{ iff } \llbracket C, \kappa_{\mathfrak{S}} \rrbracket(a) \in \mathfrak{s}\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models r(a, b) \text{ iff } r(a, b) \in \mathfrak{s}\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models \neg r(a, b) \text{ iff } \neg r(a, b) \in \mathfrak{s}\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models \mathbf{K}r(a, b) \text{ iff } b \in \kappa_{\mathfrak{S}}(a, r)\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models \neg\mathbf{K}r(a, b) \text{ iff } b \notin \kappa_{\mathfrak{S}}(a, r)\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models \mathbf{K}\neg r(a, b) \text{ iff } b \in \kappa_{\mathfrak{S}}(a, \neg r)\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models \neg\mathbf{K}\neg r(a, b) \text{ iff } b \notin \kappa_{\mathfrak{S}}(a, \neg r)\\
&(\mathfrak{s}, \mathfrak{S}) \mid\!\models C \sqsubseteq D \text{ iff } C \sqsubseteq D \in \mathfrak{s}.
\end{aligned} \tag{17}$$

For an $\mathcal{ALCOK}$-ground formula ψ build from assertions from $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ satisfaction in $(\mathfrak{s}, \mathfrak{S})$, denoted by $(\mathfrak{s}, \mathfrak{S}) \mid\models \psi$, is defined in the obvious way. Likewise, for a non-epistemic static type $\mathfrak{s} = \mathsf{s\text{-}type}(\mathcal{I})$ of an interpretation $\mathcal{I}$ and an objective ground formula ψ over assertions from $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$, satisfaction of ψ in $\mathfrak{s}$, denoted by $\mathfrak{s} \models \psi$, is defined in the obvious way.

As a consequence of Lemma 54 we get that satisfaction in $(\mathcal{I}, \mathcal{W})$ and $(\mathfrak{s}, \mathfrak{S})$ coincide.

Lemma 55. *Let $\langle(\mathcal{I}, \mathcal{W}), \rho\rangle$ be a reachable state in $T_{\mathcal{P}}$, $\mathfrak{s} = \mathsf{s\text{-}type}(\mathcal{I})$ and $\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\}$, $\varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$ an axiom and ψ a ground formula over assertions from $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. It holds that $(\mathcal{I}, \mathcal{W}) \mid\models \varrho$ iff $(\mathfrak{s}, \mathfrak{S}) \mid\models \varrho$ and $(\mathcal{I}, \mathcal{W}) \mid\models \psi$ iff $(\mathfrak{s}, \mathfrak{S}) \mid\models \psi$.*

Given the abstraction of an epistemic interpretation in terms of the static type of the external world and the static types of all possible worlds we need to determine the abstraction of the updated epistemic interpretation after an action was performed. As in [BZ13a] we use the notion of a *dynamic type* of an interpretation. The dynamic type captures also the static types of all possible updated interpretations and is defined as in [BZ13a] based on the set $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and the set of all literals given as follows:

$$\begin{aligned} Lit := &\{A(a), \neg A(a) \mid (\mathsf{eff}, \mathsf{sense}) \in \Sigma, a \in \mathsf{Ind}, \psi/(\neg)A(x) \in \mathsf{eff}\} \cup \\ &\{r(a,b), \neg r(a,b) \mid (\mathsf{eff}, \mathsf{sense}) \in \Sigma, a, b \in \mathsf{Ind}, \psi/(\neg)r(x,y) \in \mathsf{eff}\}. \end{aligned}$$

Definition 56 (dynamic types)**.** Let $\mathcal{P}$ be a program and $\mathcal{I}$ an interpretation. The *dynamic type* of $\mathcal{I}$ w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ is defined as follows:

$$\mathsf{d\text{-}type}(\mathcal{I}) := \{(\varrho, \mathsf{L}) \in \widehat{\mathcal{C}}(\mathcal{P}, c_u) \times 2^{Lit} \mid \mathcal{I}^{\mathsf{L}} \models \varrho\}.$$

▲

Thus, a dynamic type $\mathfrak{d}$ of an interpretation is a set $\mathfrak{d} \subseteq \widehat{\mathcal{C}}(\mathcal{P}, c_u) \times 2^{Lit}$ such that for each $\varrho \in \widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and each $\mathsf{L} \in 2^{Lit}$ it holds that either $(\varrho, \mathsf{L}) \in \mathfrak{d}$ or $(\neg\varrho, \mathsf{L}) \in \mathfrak{d}$.

Let $\mathfrak{d} = \mathsf{d\text{-}type}(\mathcal{I})$ be the dynamic type of an interpretation $\mathcal{I}$ and $\mathsf{L} \in 2^{Lit}$ a set of literals. The *static type of* $(\mathfrak{d}, \mathsf{L})$ is defined by

$$\mathsf{s\text{-}type}(\mathfrak{d}, \mathsf{L}) := \{\varrho \mid (\varrho, \mathsf{L}) \in \mathfrak{d}\}.$$

Obviously, it holds that

$$\mathsf{s\text{-}type}(\mathsf{d\text{-}type}(\mathcal{I}), \mathsf{L}) = \mathsf{s\text{-}type}(\mathcal{I}^{\mathsf{L}}) \tag{18}$$

for any interpretation $\mathcal{I}$ and $\mathsf{L} \in 2^{Lit}$.

The *set of all dynamic types* w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ is denoted by

$$\mathfrak{D} = \{\mathsf{d\text{-}type}(\mathcal{I}) \mid \mathcal{I} \in \mathcal{M}(\mathcal{K})\}.$$

Let $\mathcal{W}$ be a knowledge state reachable in $T_{\mathcal{P}}$. For each $\mathcal{J} \in \mathcal{W}$ there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $\mathsf{L} \in 2^{Lit}$ such that $\mathcal{J} = \mathcal{I}^{\mathsf{L}}$. For the construction of the finite abstract transition system we use the pair $(\mathsf{d\text{-}type}(\mathcal{I}), \mathsf{L})$ as abstraction of $\mathcal{J}$, where L are the current accumulated physical effects and $\mathsf{d\text{-}type}(\mathcal{I})$ already encodes the static types of all future evolutions of $\mathcal{I}$. Thus, the abstraction of an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ occurring in a state of $T_{\mathcal{P}}$ is a pair $(\mathfrak{t}, \mathfrak{T})$ with $\mathfrak{T} \subseteq \mathfrak{D} \times 2^{Lit}$ and $\mathfrak{t} \in \mathfrak{T}$. We define a simulation relation between $(\mathcal{I}, \mathcal{W})$ and $(\mathfrak{t}, \mathfrak{T})$.

Definition 57. Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be a program, $\mathfrak{D}$ the set of all dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and *Lit* as defined above. Let $(\mathcal{I}, \mathcal{W})$ be an epistemic interpretation and $(\mathfrak{t}, \mathfrak{T})$ a pair with $\mathfrak{T} \subseteq \mathfrak{D} \times 2^{Lit}$ and $\mathfrak{t} \in \mathfrak{T}$. It holds that $(\mathcal{I}, \mathcal{W}) \simeq (\mathfrak{t}, \mathfrak{T})$ iff the following conditions are satisfied:

1. There exists $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$ and $\mathsf{L} \in 2^{Lit}$ such that $\mathfrak{t} = (\mathsf{d\text{-}type}(\mathcal{I}_0), \mathsf{L})$ and $\mathcal{I} = {\mathcal{I}_0}^{\mathsf{L}}$.
2. For each $\mathcal{J} \in \mathcal{W}$ there exists $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ and $\mathsf{L} \in 2^{Lit}$ such that $\mathcal{J} = {\mathcal{J}_0}^{\mathsf{L}}$ and $(\mathsf{d\text{-}type}(\mathcal{J}_0), \mathsf{L}) \in \mathfrak{T}$.
3. For each $(\mathfrak{d}, \mathsf{L}) \in \mathfrak{T}$ there exists $\mathcal{J}_0 \in \mathcal{M}(\mathcal{K})$ such that $\mathfrak{d} = \mathsf{d\text{-}type}(\mathcal{J}_0)$ and ${\mathcal{J}_0}^{\mathsf{L}} \in \mathcal{W}$.

▲

Next, we lift the transition relation on epistemic interpretations induced by a primitive ground action to the abstract level of types. Let $\mathfrak{t}, \mathfrak{t}' \in \mathfrak{D} \times 2^{Lit}$ and $\alpha = (\mathsf{eff}, \mathsf{sense}) \in \Sigma_{\mathsf{g}}$. We define

$$\widehat{\mathcal{E}}(\mathsf{eff}, \mathfrak{t}) := \{\gamma \mid \psi/\gamma \in \mathsf{eff}, \mathsf{s\text{-}type}(\mathfrak{t}) \models \psi\}.$$

And it holds that $\mathfrak{t}$ and $\mathfrak{t}'$ are *sensing compatible* w.r.t. α, written as $\mathfrak{t} \mathrel{\widehat{\sim}_\alpha} \mathfrak{t}'$, iff for all $\psi \in \mathsf{sense}$ it holds that $\mathsf{s\text{-}type}(\mathfrak{t}) \models \psi$ iff $\mathsf{s\text{-}type}(\mathfrak{t}') \models \psi$. Let $\mathfrak{T}, \mathfrak{T}' \subseteq \mathfrak{D} \times 2^{Lit}$ with $\mathfrak{t} = (\mathfrak{d}, \mathsf{L})$. It holds that $(\mathfrak{t}, \mathfrak{T}) \Longrightarrow_\alpha (\mathfrak{t}', \mathfrak{T}')$ iff the following conditions are satisfied:

- $\mathfrak{t}' = (\mathfrak{d}, \mathsf{L} \setminus \neg\mathsf{L}' \cup \mathsf{L}')$ with $\mathsf{L}' = \widehat{\mathcal{E}}(\mathsf{eff}, \mathfrak{t})$ and
- $\mathfrak{T}' = \{(\mathfrak{d}', \mathsf{L} \setminus \neg\mathsf{L}' \cup \mathsf{L}') \mid (\mathfrak{d}', \mathsf{L}) \in \mathfrak{T}, \mathfrak{t} \mathrel{\widehat{\sim}_\alpha} (\mathfrak{d}', \mathsf{L}), \mathsf{L}' = \widehat{\mathcal{E}}(\mathsf{eff}, (\mathfrak{d}', \mathsf{L}))\}$.

We show that executing actions preserves the simulation relation.

Lemma 58. *Let $\langle(\mathcal{I}_0, \mathcal{W}_0), \rho\rangle$ be a reachable state in $T_{\mathcal{P}}$, $\mathfrak{D}$ the set of all dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$, $\mathfrak{T}_0 \subseteq \mathfrak{D} \times 2^{Lit}$, $\mathfrak{t}_0 \in \mathfrak{T}_0$ such that $(\mathcal{I}_0, \mathcal{W}_0) \simeq (\mathfrak{t}_0, \mathfrak{T}_0)$ and $\alpha = (\mathsf{eff}, \mathsf{sense}) \in \Sigma_{\mathsf{g}}$. For $(\mathcal{I}_1, \mathcal{W}_1)$ and $(\mathfrak{t}_1, \mathfrak{T}_1)$ with $(\mathcal{I}_0, \mathcal{W}_0) \Longrightarrow_\alpha (\mathcal{I}_1, \mathcal{W}_1)$ and $(\mathfrak{t}_0, \mathfrak{T}_0) \Longrightarrow_\alpha (\mathfrak{t}_1, \mathfrak{T}_1)$, respectively, it holds that $(\mathcal{I}_1, \mathcal{W}_1) \simeq (\mathfrak{t}_1, \mathfrak{T}_1)$.*

Proof. $(\mathcal{I}_0, \mathcal{W}_0) \simeq (\mathfrak{t}_0, \mathfrak{T}_0)$ implies that there exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $\mathsf{L} \in 2^{Lit}$ such that $\mathcal{I}_0 = \mathcal{I}^{\mathsf{L}}$ and $\mathfrak{t}_0 = (\mathsf{d\text{-}type}(\mathcal{I}), \mathsf{L})$. It follows that $\mathsf{s\text{-}type}(\mathcal{I}_0) = \mathsf{s\text{-}type}(\mathfrak{t}_0)$ with (18). Let $\psi/\gamma \in \mathsf{eff}$. By definition ψ is a boolean combination of objective assertions from $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$. Using Lemma 55 it follows that $\mathcal{I}_0 \models \psi$ iff $\mathsf{s\text{-}type}(\mathcal{I}_0) \models \psi$ iff $\mathsf{s\text{-}type}(\mathfrak{t}_0) \models \psi$. It follows that $\mathcal{E}(\mathcal{I}_0, \mathsf{eff}) = \widehat{\mathcal{E}}(\mathfrak{t}_0, \mathsf{eff})$. By definition of the transition relation and with $\mathfrak{t}_0 = (\mathsf{d\text{-}type}(\mathcal{I}), \mathsf{L})$ we have

$$\mathfrak{t}_1 = (\mathsf{d\text{-}type}(\mathcal{I}), \mathsf{L} \setminus \neg\widehat{\mathcal{E}}(\mathfrak{t}_0, \mathsf{eff}) \cup \widehat{\mathcal{E}}(\mathfrak{t}_0, \mathsf{eff}))$$

and with Lemma 2

$$\mathcal{I}_1 = {\mathcal{I}_0}^{\mathcal{E}(\mathcal{I}_0, \mathsf{eff})} = {\mathcal{I}^{\mathsf{L}}}^{\mathcal{E}(\mathcal{I}_0, \mathsf{eff})} = \mathcal{I}^{\mathsf{L} \setminus \neg\mathcal{E}(\mathcal{I}_0, \mathsf{eff}) \cup \mathcal{E}(\mathcal{I}_0, \mathsf{eff})}.$$

Therefore, $\mathcal{I}_1$ and $\mathfrak{t}_1$ satisfy the first condition of $\simeq$.

Next, we show that the second condition is also satisfied. Let $\mathcal{J}_1 \in \mathcal{W}_1$. There exists $\mathcal{J}_0 \in \mathcal{W}_0$ with $\mathcal{I}_0 \sim_\alpha \mathcal{J}_0$ and $\mathcal{J}_1 = {\mathcal{J}_0}^{\mathcal{E}(\mathcal{J}_0, \mathsf{eff})}$. Since by assumption $(\mathcal{I}_0, \mathcal{W}_0) \simeq (\mathfrak{t}_0, \mathfrak{T}_0)$, there is $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ and $\mathsf{L} \in 2^{Lit}$ such that $\mathcal{J}_0 = \mathcal{J}^{\mathsf{L}}$ and $(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L}) \in \mathfrak{T}_0$. It follows that $\mathsf{s\text{-}type}(\mathcal{J}_0) = \mathsf{s\text{-}type}(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L})$. Let $\psi \in \mathsf{sense}$. By definition ψ is a boolean combination of objective assertions from $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$. It follows that $\mathsf{s\text{-}type}(\mathfrak{t}_0) \models \psi$

iff $\mathsf{s\text{-}type}(\mathcal{I}_0) \models \psi$ (since $\mathsf{s\text{-}type}(\mathcal{I}_0) = \mathsf{s\text{-}type}(\mathfrak{t}_0)$)

iff $\mathcal{I}_0 \models \psi$

iff $\mathcal{J}_0 \models \psi$ (since $\mathcal{I}_0 \sim_\alpha \mathcal{J}_0$)

iff $\mathsf{s\text{-}type}(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L}) \models \psi$. (since $\mathsf{s\text{-}type}(\mathcal{J}_0) = \mathsf{s\text{-}type}(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L})$)

It follows that $\mathfrak{t}_0 \mathrel{\widehat{\sim}_\alpha} (\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L})$. Let $\mathsf{L}' := \widehat{\mathcal{E}}((\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L}), \mathsf{eff})$. By definition of the transition relation it holds that

$$(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L} \setminus \neg\mathsf{L}' \cup \mathsf{L}') \in \mathfrak{T}_1.$$

From $\mathsf{s\text{-}type}(\mathcal{J}_0) = \mathsf{s\text{-}type}(\mathsf{d\text{-}type}(\mathcal{J}), \mathsf{L})$ it follows that $\mathcal{E}(\mathcal{J}_0, \mathsf{eff}) = \mathsf{L}'$. We obtain

$$\mathcal{J}_1 = \mathcal{J}_0{}^{\mathcal{E}(\mathcal{J}_0,\mathsf{eff})} = \mathcal{J}^{\mathsf{L}^{\mathcal{E}(\mathcal{J}_0,\mathsf{eff})}} = \left(\mathcal{J}^{\mathsf{L}}\right)^{\mathsf{L}'} = \mathcal{J}^{\mathsf{L}\setminus\neg\mathsf{L}'\cup\mathsf{L}'}.$$

Therefore, the second condition of $\simeq$ is satisfied. The proof the third condition works using the same arguments in the other direction. □

Now we are ready to define the abstract transition system.

In the *finite* abstract transition system of a restricted program $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ a state is of the form $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$ with $\mathfrak{T} \subseteq \mathfrak{D} \times 2^{Lit}$, $\mathfrak{t} \in \mathfrak{T}$ and $\rho \in \mathsf{subg}(\delta)$.

Furthermore, we define *executability of an atomic program* $\mathfrak{a} \in \mathsf{head}(\rho)$ in a state $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$. Let $\mathfrak{s} = \mathsf{s\text{-}type}(\mathfrak{t})$ and $\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathfrak{t}') \mid \mathfrak{t}' \in \mathfrak{T}\}$. $\mathfrak{a} \in \mathsf{head}(\rho)$ is executable in $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$ iff

- $\mathfrak{a}$ is a primitive action or
- $\mathfrak{a}$ is a test of the form $\psi?$ and $(\mathfrak{s}, \mathfrak{S}) \mathrel{|}\!\models \psi$ or
- $\mathfrak{a}$ is of the form $\mathsf{pick}(\vec{x}) : \psi?$ and there exists a variable mapping ν with $\nu(x) \in \mathsf{Ind}$ for all $x \in \mathsf{Var}(\psi)$ such that $(\mathfrak{s}, \mathfrak{S}) \mathrel{|}\!\models \psi^\nu$.

The state $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$ is called *failure state* iff no atomic program in $\mathsf{head}(\rho)$ is executable.

Definition 59 (abstract transition system)**.** Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be a restricted program and $\mathfrak{D}$ the set of all dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$. The *abstract transition system* $\widehat{T}_{\mathcal{P}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$ induced by $\mathcal{P}$ consists of the set of states

$$\widehat{Q} := \{\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle \mid \mathfrak{T} \subseteq \mathfrak{D} \times 2^{Lit}, \mathfrak{t} \in \mathfrak{T}, \rho \in \mathsf{subg}(\delta)\},$$

a set of initial state $\widehat{I} \subseteq \widehat{Q}$ given by

$$\widehat{I} := \{\langle(\mathfrak{t}, \mathfrak{D} \times \{\emptyset\}), \delta\rangle \mid \mathfrak{t} \in \mathfrak{D} \times \{\emptyset\}\}$$

and a labeled transition relation on $\widehat{Q}$ that is defined as follows: Let $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle \in \widehat{Q}$ and $\langle(\mathfrak{t}', \mathfrak{T}'), \rho'\rangle \in \widehat{Q}$ be two states and $\mathfrak{s} = \mathsf{s\text{-}type}(\mathfrak{t})$ and $\mathfrak{S} = \{\mathsf{s\text{-}type}(\mathfrak{t}'') \mid \mathfrak{t}'' \in \mathfrak{T}\}$. It holds that

$$\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle \overset{\mathfrak{a}}{\twoheadrightarrow} \langle(\mathfrak{t}', \mathfrak{T}'), \rho'\rangle$$

iff one of the following conditions is satisfied:

1. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a} \in \Sigma_{\mathsf{g}}$ such that $\mathfrak{a} \neq \mathfrak{f}$, $(\mathfrak{t}, \mathfrak{T}) \Longrightarrow_{\mathfrak{a}} (\mathfrak{t}', \mathfrak{T}')$ and $\rho' \in \mathsf{tail}(\mathfrak{a}, \rho)$.
2. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a}$ is a test of the form $\psi?$, $(\mathfrak{s}, \mathfrak{S}) \mathrel{|}\!\models \psi$, $(\mathfrak{t}', \mathfrak{T}') = (\mathfrak{t}, \mathfrak{T})$ and $\rho' \in \mathsf{tail}(\psi?, \rho)$.
3. $\mathfrak{a} \in \mathsf{head}(\rho)$, $\mathfrak{a}$ is a pick of the form $\mathsf{pick}(\vec{x}) : \psi?$, there exists a variable mapping ν such that $(\mathfrak{s}, \mathfrak{S}) \mathrel{|}\!\models \psi^\nu$ and there exists $\zeta \in \mathsf{tail}(\mathsf{pick}(\vec{x}) : \psi?, \rho)$ such that $\rho' = \zeta^\nu$, and $(\mathfrak{t}', \mathfrak{T}') = (\mathfrak{t}, \mathfrak{T})$.
4. $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$ is a failure state, $\mathfrak{a} = \mathfrak{f}$, $(\mathfrak{t}, \mathfrak{T}) \Longrightarrow_{\mathfrak{f}} (\mathfrak{t}', \mathfrak{T}')$ and $\rho' = \rho$.

▲

We define a binary relation on the states of $T_{\mathcal{P}} = (Q, \rightarrow, I)$ and $\widehat{T}_{\mathcal{P}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$. Let $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \in Q$ and $\langle(\mathsf{t},\mathfrak{T}),\zeta\rangle \in \widehat{Q}$. We write $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \bowtie \langle(\mathsf{t},\mathfrak{T}),\zeta\rangle$ iff $(\mathcal{I},\mathcal{W}) \simeq (\mathsf{t},\mathfrak{T})$ and $\rho = \zeta$.

We proof the relation of the concrete transition system with the abstract one.

Lemma 60. *Let $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$ and $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$ be two reachable states in $T_{\mathcal{P}}$ and $\widehat{T}_{\mathcal{P}}$, respectively such that $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \bowtie \langle(\mathsf{t},\mathfrak{T}),\rho\rangle$.*

1. *If there is a transition $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \xrightarrow{\mathfrak{a}} \langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle$ in $T_{\mathcal{P}}$, then there is a transition $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle \overset{\mathfrak{a}}{\twoheadrightarrow} \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$ in $\widehat{T}_{\mathcal{P}}$ such that $\langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle \bowtie \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$.*

2. *If there is a transition $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle \overset{\mathfrak{a}}{\twoheadrightarrow} \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$ in $\widehat{T}_{\mathcal{P}}$, then there is a transition $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \xrightarrow{\mathfrak{a}} \langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle$ in $T_{\mathcal{P}}$ such that $\langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle \bowtie \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$.*

Proof. Assume that $\mathfrak{a} \in \Sigma_{\mathsf{g}}$. With Lemma 58 it follows that $(\mathcal{I}',\mathcal{W}') \simeq (\mathsf{t}',\mathfrak{T}')$ with $(\mathcal{I},\mathcal{W}) \Longrightarrow_{\mathfrak{a}} (\mathcal{I}',\mathcal{W}')$ and $(\mathsf{t},\mathfrak{T}) \Longrightarrow_{\mathfrak{a}} (\mathsf{t}',\mathfrak{T}')$. With $\rho' \in \mathsf{tail}(\mathfrak{a},\rho)$ it follows that $\langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle \bowtie \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$ with $\langle(\mathcal{I},\mathcal{W}),\rho\rangle \xrightarrow{\mathfrak{a}} \langle(\mathcal{I}',\mathcal{W}'),\rho'\rangle$ and $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle \overset{\mathfrak{a}}{\twoheadrightarrow} \langle(\mathsf{t}',\mathfrak{T}'),\rho'\rangle$.

Now, assume that $\mathfrak{a} \in \mathsf{head}(\rho)$ is a test or a guarded pick. By assumption it holds that $(\mathcal{I},\mathcal{W}) \simeq (\mathsf{t},\mathfrak{T})$. It follows that

$$\mathfrak{s} := \mathsf{s\text{-}type}(\mathcal{I}) = \mathsf{s\text{-}type}(\mathsf{t}) \text{ and } \mathfrak{S} := \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}\} = \{\mathsf{s\text{-}type}(\hat{\mathsf{t}}) \mid \hat{\mathsf{t}} \in \mathfrak{T}\}. \qquad (19)$$

Lemma 48 implies that all individuals occurring in $\mathfrak{a}$ are contained in Ind. Assume that $\mathfrak{a} = \psi?$ is a test. We know that ψ is a formula built from assertions contained in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. Lemma 55 and (19) implies that $\mathfrak{a}$ is executable in $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$ iff it is executable in $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$.

Assume that $\mathfrak{a} = \mathsf{pick}(\vec{x}) : \psi?$ is a guarded pick. It holds that $\mathfrak{a}$ is executable in $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$

iff there exists a variable mapping ν such that $(\mathcal{I},\mathcal{W}) \mid\models \psi^{\nu}$

iff there exists a variable mapping ν such that $\nu(x) \in \mathsf{Ind}$ for all $x \in \mathsf{Var}(\psi)$ and $(\mathcal{I},\mathcal{W}) \mid\models \psi^{\nu}$ (by Lemma 48)

iff there exists a variable mapping ν such that $\nu(x) \in \mathsf{Ind}$ for all $x \in \mathsf{Var}(\psi)$ and $(\mathfrak{s},\mathfrak{S}) \mid\models \psi^{\nu}$ (by Lemma 55 and (19))

iff $\mathfrak{a}$ is executable in $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$.

Thus, it follows that $\mathfrak{a} \in \mathsf{head}(\rho)$ is executable in $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$ iff it is executable in $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$. Consequently, it also follows that $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$ is a failure state iff $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$ is one. It easy to see that if a test or guarded pick is executed in $\langle(\mathcal{I},\mathcal{W}),\rho\rangle$ and in $\langle(\mathsf{t},\mathfrak{T}),\rho\rangle$ with the same variable mapping, then the resulting states are also in $\bowtie$-relation. □

We now lift the $\bowtie$-relation to runs as follows: Let π be a run in $T_{\mathcal{P}}$ of the form

$$\pi = \langle(\mathcal{I}_0,\mathcal{W}_0),\delta_0\rangle \xrightarrow{\mathfrak{a}_0} \langle(\mathcal{I}_1,\mathcal{W}_1),\delta_1\rangle \xrightarrow{\mathfrak{a}_1} \langle(\mathcal{I}_2,\mathcal{W}_2),\delta_2\rangle \xrightarrow{\mathfrak{a}_2} \cdots$$

and $\hat{\pi}$ an infinite path in $\widehat{T}_{\mathcal{P}}$ starting in an initial state of the form

$$\hat{\pi} = \langle(\mathsf{t}_0,\mathfrak{T}_0),\rho_0\rangle \overset{\mathfrak{a}_0}{\twoheadrightarrow} \langle(\mathsf{t}_1,\mathfrak{T}_1),\rho_1\rangle \overset{\mathfrak{a}_1}{\twoheadrightarrow} \langle(\mathsf{t}_2,\mathfrak{T}_2),\rho_2\rangle \overset{\mathfrak{a}_2}{\twoheadrightarrow} \cdots$$

We write $\pi \bowtie \hat{\pi}$ iff for all $i = 0, 1, 2, \ldots$ it holds that $\langle(\mathcal{I}_i,\mathcal{W}_i),\delta_i\rangle \bowtie \langle(\mathsf{t}_i,\mathfrak{T}_i),\rho_i\rangle$.

Lemma 61. *Let $T_{\mathcal{P}}$ be the transition system of a restricted program $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ and $\widehat{T}_{\mathcal{P}}$ the corresponding abstract transition system. For each run π in $T_{\mathcal{P}}$ there exists an infinite path $\hat{\pi}$ in $\widehat{T}_{\mathcal{P}}$ starting in an initial state such that $\pi \bowtie \hat{\pi}$ and for each infinite path $\hat{\pi}'$ in $\widehat{T}_{\mathcal{P}}$ starting in an initial state there exists a run π' in $T_{\mathcal{P}}$ such that $\pi' \bowtie \hat{\pi}'$.*

Proof. Let $\pi = s_0 s_1 s_2 \cdots$ be a run in $T_{\mathcal{P}} = (Q, \rightarrow, I)$ such that $s_i = \langle(\mathcal{I}_i, \mathcal{W}_i), \delta_i\rangle$ and $s_i \xrightarrow{\mathfrak{a}_i} s_{i+1}$ for all $i \in \mathbb{N}$. We inductively define a corresponding path $\hat{\pi} = \hat{s}_0 \hat{s_1} \hat{s_2} \cdots$ in $\widehat{T}_{\mathcal{P}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$ such that $\hat{s}_0 \in \widehat{I}$, $\hat{s}_i \overset{\mathfrak{a}_i}{\twoheadrightarrow} \hat{s}_{i+1}$ and $s_i \bowtie \hat{s}_i$ for all $i \in \mathbb{N}$.

It holds that $\mathcal{W}_0 = \mathcal{M}(\mathcal{K})$ and $\delta_0 = \delta$. We define $\hat{s}_0 := \langle(\mathsf{t}_0, \mathfrak{T}_0), \delta\rangle$ with $\mathsf{t}_0 = (\mathsf{d\text{-}type}(\mathcal{I}_0), \emptyset)$ and $\mathfrak{T}_0 = \mathfrak{D} \times \{\emptyset\}$. Clearly, $\hat{s}_0 \in \widehat{I}$ and $s_0 \bowtie \hat{s}_0$.

Let $n > 0$ and assume that states $\hat{s}_0 \hat{s}_1 \cdots \hat{s}_n$ are already defined such that for all $j = 0, \ldots, n-1$ it holds that $\hat{s}_j \overset{\mathfrak{a}_j}{\twoheadrightarrow} \hat{s}_{j+1}$ and $s_j \bowtie \hat{s}_j$ and $s_n \bowtie \hat{s}_n$. By assumption there is a transition $s_n \xrightarrow{\mathfrak{a}_n} s_{n+1}$ in the concrete transition system and it holds that $s_n \bowtie \hat{s}_n$. Lemma 60 now implies that we can choose a state $\hat{s}_{n+1} \in \widehat{Q}$ such that $\hat{s}_n \overset{\mathfrak{a}_n}{\twoheadrightarrow} \hat{s}_{n+1}$ and $s_{n+1} \bowtie \hat{s}_{n+1}$.

For the other direction consider a path in $\widehat{T}_{\mathcal{P}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$ starting in an initial state of the form $\hat{\pi} = \hat{s}_0 \hat{s}_1 \hat{s}_2 \cdots$ with $\hat{s}_i = \langle(\mathsf{t}_i, \mathfrak{T}_i), \delta_i\rangle$ and $\hat{s}_i \overset{\mathfrak{a}_i}{\twoheadrightarrow} \hat{s}_{i+1}$ for all $i \in \mathbb{N}$ and $\hat{s}_0 \in \widehat{I}$. As before, we inductively define a run $\pi = s_0 s_1 s_2 \cdots$ in $T_{\mathcal{P}} = (Q, \rightarrow, I)$ such that $\pi \bowtie \hat{\pi}$.

It holds that $\hat{s}_0 = \langle(\mathsf{t}_0, \mathfrak{T}_0), \delta\rangle$ with $\mathfrak{T}_0 = \mathfrak{D} \times \{\emptyset\}$ where $\mathfrak{D}$ is the set of all dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and $\mathsf{t}_0 = (\mathsf{d\text{-}type}(\mathcal{I}_0), \emptyset)$ for some $\mathcal{I}_0 \in \mathcal{M}(\mathcal{K})$. We define $s_0 := \langle(\mathcal{I}_0, \mathcal{M}(\mathcal{K})), \delta\rangle$. Obviously, $s_0 \in I$ and $s_0 \bowtie \hat{s}_0$.

Let $n > 0$ and assume that states $s_0 s_1 \cdots s_n$ are already defined such that for all $j = 0, \ldots, n-1$ it holds that $s_j \xrightarrow{\mathfrak{a}_j} s_{j+1}$, $s_j \bowtie \hat{s}_j$ and $s_n \bowtie \hat{s}_n$. Consider the transition $\hat{s}_n \overset{\mathfrak{a}_n}{\twoheadrightarrow} \hat{s}_{n+1}$ in $\hat{\pi}$. Since $s_n \bowtie \hat{s}_n$ and Lemma 60 holds, we can choose a state $s_{n+1} \in Q$ such that $s_n \xrightarrow{\mathfrak{a}_n} s_{n+1}$ and $s_{n+1} \bowtie \hat{s}_{n+1}$. □

Now, we can use the finite abstract transition system for model checking. We first build the *propositional abstraction* of the $\mathcal{ALCOK}$-LTL formula Φ that we want to verify in $\mathcal{P}$. We introduce for each axiom in the context $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$ an atomic proposition. Let

$$\mathsf{AP} := \{p_\varrho \mid \varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})\}.$$

be the finite set of all relevant atomic propositions. For convenience we assume that all axioms in Φ are taken from $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. The propositional abstraction of Φ, denoted by $\widehat{\Phi}$, is a propositional LTL-formula that is obtained from Φ by replacing each axiom ϱ in Φ by the corresponding atomic proposition p_ϱ. The semantics of LTL is defined in terms of a *propositional LTL-structure* which is an infinite sequence of the form $\mathfrak{L} = X_0 X_1 X_2 \ldots$ with $X_i \in 2^{\mathsf{AP}}$ for all $i = 0, 1, 2, \ldots$ Satisfaction of an LTL-formula in an LTL-structure $\mathfrak{L}$ at a given time point is defined in the usual way.

We introduce a labeling function $L_{\mathsf{AP}} : \widehat{Q} \to 2^{\mathsf{AP}}$ for the abstract states defined as follows: Let $\hat{s} = \langle(\mathsf{t}, \mathfrak{T}), \rho\rangle \in \widehat{Q}$ be a state in $\widehat{T}_{\mathcal{P}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$ and $\mathfrak{s} := \mathsf{s\text{-}type}(\mathsf{t})$ and $\mathfrak{S} := \{\mathsf{s\text{-}type}(\mathsf{t}') \mid \mathsf{t}' \in \mathfrak{T}\}$.

$$L_{\mathsf{AP}}(\hat{s}) := \{p_\varrho \in \mathsf{AP} \mid (\mathfrak{s}, \mathfrak{S}) \mathrel{|}\!\models \varrho\}.$$

The propositional LTL-structure for an infinite path $\hat{\pi} = \hat{s}_0 \hat{s}_1 \hat{s}_2 \cdots$ in $\widehat{T}_{\mathcal{P}}$ with $\hat{s}_0 \in \widehat{I}$ is given as follows:

$$\mathfrak{L}(\hat{\pi}) = L_{\mathsf{AP}}(\hat{s}_0) L_{\mathsf{AP}}(\hat{s}_1) L_{\mathsf{AP}}(\hat{s}_2) \cdots$$

Lemma 62. *Let $\mathcal{P}$ be a restricted program, Φ an $\mathcal{ALCOK}$-LTL formula over axioms in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$, π a run in $T_{\mathcal{P}}$ and $\hat{\pi}$ an infinite path in $\widehat{T}_{\mathcal{P}}$ such that $\pi \bowtie \hat{\pi}$. For all time points $i \in \mathbb{N}$ it holds that $\mathfrak{I}(\pi), i \models \Phi$ iff $\mathfrak{L}(\hat{\pi}), i \models \widehat{\Phi}$.*

Proof. The claim is shown by induction over the structure of Φ. Let $\pi = s_0 s_1 s_2 \cdots$ with $s_i = \langle(\mathcal{I}_i, \mathcal{W}_i), \delta_i\rangle$, $\hat{\pi} = \hat{s}_0 \hat{s}_1 \hat{s}_2 \cdots$ with $\hat{s}_i = \langle(\mathsf{t}_i, \mathfrak{T}_i), \delta_i\rangle$ and $s_i \bowtie \hat{s}_i$ for all $i = 0, 1, 2, \ldots$

$\Phi = \varrho$: We have $\varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})$. Let i be a time point. It holds that $\mathfrak{I}(\pi), i \models \varrho$

iff $(\mathcal{I}_i, \mathcal{W}_i) \mid\!\models \varrho$

iff $(\mathfrak{s}_i, \mathfrak{S}_i) \mid\!\models \varrho$ with $\mathfrak{s}_i = \mathsf{s\text{-}type}(\mathcal{I}_i)$ and $\mathfrak{S}_i = \{\mathsf{s\text{-}type}(\mathcal{J}) \mid \mathcal{J} \in \mathcal{W}_i\}$. (by Lemma 55)

iff $(\mathfrak{s}'_i, \mathfrak{S}'_i) \mid\!\models \varrho$ with $\mathfrak{s}'_i = \mathsf{s\text{-}type}(\mathsf{t}_i)$ and $\mathfrak{S}'_i = \{\mathsf{s\text{-}type}(\mathsf{t}') \mid \mathsf{t}' \in \mathfrak{T}_i\}$ since $s_i \bowtie \hat{s}_i$ implies $(\mathcal{I}_i, \mathcal{W}_i) \simeq (\mathsf{t}_i, \mathfrak{T}_i)$ which implies $\mathfrak{s}_i = \mathfrak{s}'_i$ and $\mathfrak{S}_i = \mathfrak{S}'_i$.

iff $p_\varrho \in L_{\mathsf{AP}}(\langle(\mathsf{t}_i, \mathfrak{T}_i), \delta_i\rangle)$

iff $\mathfrak{L}(\hat{\pi}), i \models p_\varrho$.

$\Phi = \neg\Phi'$: By induction we assume that $\mathfrak{I}(\pi), j \models \Phi'$ iff $\mathfrak{L}(\hat{\pi}), j \models \widehat{\Phi'}$ for all $j = 0, 1, 2, \ldots$

Therefore, $\mathfrak{I}(\pi), i \models \neg\Phi'$ iff $\mathfrak{I}(\pi), i \not\models \Phi'$ iff $\mathfrak{L}(\hat{\pi}), i \not\models \widehat{\Phi'}$ iff $\mathfrak{L}(\hat{\pi}), i \models \widehat{\neg\Phi'}$.

$\Phi = \Phi_1 \wedge \Phi_2$: By induction we assume that $\mathfrak{I}(\pi), j \models \Phi_z$ iff $\mathfrak{L}(\hat{\pi}), j \models \widehat{\Phi_z}$ for all $j = 0, 1, 2, \ldots$ and $z \in \{0, 1\}$.

Therefore $\mathfrak{I}(\pi), i \models \Phi_1 \wedge \Phi_2$ iff $\mathfrak{L}(\hat{\pi}), i \models \widehat{\Phi_1 \wedge \Phi_2}$.

$\Phi = \mathsf{X}\Phi'$: By induction we assume that $\mathfrak{I}(\pi), j \models \Phi'$ iff $\mathfrak{L}(\hat{\pi}), j \models \widehat{\Phi'}$ for all $j = 0, 1, 2, \ldots$

Therefore, $\mathfrak{I}(\pi), i \models \mathsf{X}\Phi'$ iff $\mathfrak{I}(\pi), i+1 \models \Phi'$ iff $\mathfrak{L}(\hat{\pi}), i+1 \models \widehat{\Phi'}$ iff $\mathfrak{L}(\hat{\pi}), i \models \widehat{\mathsf{X}\Phi'}$.

$\Phi = \Phi_1 \,\mathsf{U}\, \Phi_2$: By induction we assume that $\mathfrak{I}(\pi), n \models \Phi_z$ iff $\mathfrak{L}(\hat{\pi}), n \models \widehat{\Phi_z}$ for all $n = 0, 1, 2, \ldots$ and $z \in \{0, 1\}$.

Therefore, $\mathfrak{I}(\pi), i \models \Phi_1 \,\mathsf{U}\, \Phi_2$ iff $\exists k \geq i : \mathfrak{I}(\pi), k \models \Phi_2$ and $\forall j, i \leq j < k : \mathfrak{I}(\pi), j \models \Phi_1$ iff $\exists k \geq i : \mathfrak{L}(\hat{\pi}), k \models \widehat{\Phi_2}$ and $\forall j, i \leq j < k : \mathfrak{L}(\hat{\pi}), j \models \widehat{\Phi_1}$ iff $\mathfrak{L}(\hat{\pi}), i \models \widehat{\Phi_1 \,\mathsf{U}\, \Phi_2}$.

□

From Lemma 61 and 62 it follows that the verification problem for a restricted program $\mathcal{P}$ and an $\mathcal{ALCOK}$-LTL formula Φ reduces to a propositional LTL model checking problem for $\widehat{T}_{\mathcal{P}}$ and $\widehat{\Phi}$. To decide the verification problem we need to construct $\widehat{T}_{\mathcal{P}}$. The essential part is to compute the set of all dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ denoted by $\mathfrak{D}$. The objective axioms of the knowledge closure contained in $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ can be obtained from $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. To compute $\mathfrak{D}$ we first enumerate all finitely many sets $\mathfrak{d} \subseteq \widehat{\mathcal{C}}(\mathcal{P}, c_u) \times 2^{Lit}$ such that for each $\varrho \in \widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and each $\mathsf{L} \in 2^{Lit}$ either $(\varrho, \mathsf{L}) \in \mathfrak{d}$ or $(\neg\varrho, \mathsf{L}) \in \mathfrak{d}$. Then we need to check for each such set $\mathfrak{d}$ whether a model $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ exists such that $\mathfrak{d} = \mathsf{d\text{-}type}(\mathcal{I})$. For this type checking task we use the same idea as for solving the projection problem. For a given complete set $\mathfrak{d} \subseteq \widehat{\mathcal{C}}(\mathcal{P}, c_u) \times 2^{Lit}$ we construct a knowledge base $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ such that $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ has a model iff $\mathfrak{d} = \mathsf{d\text{-}type}(\mathcal{I})$ for some $\mathcal{I} \in \mathcal{M}(\mathcal{K})$. Let $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ and $2^{Lit} = \{\mathsf{L}_0, \ldots, \mathsf{L}_m\}$. The idea of constructing $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ is to encode the set of models $\{\mathcal{I}^{\mathsf{L}_0}, \mathcal{I}^{\mathsf{L}_1}, \ldots, \mathcal{I}^{\mathsf{L}_m}\}$ into a single model of $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$. However, we first need to deal with the negated $\mathcal{ALC}$-CIs of the form $\neg(C \sqsubseteq D)$ contained in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. We introduce additional concept assertions as witnesses for a violation of a CI.

Let $\mathcal{G} \subseteq \widehat{\mathcal{C}}(\mathcal{P}, c_u)$ be the set of all CIs contained in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$. Let $\mathcal{G} := \{C_0 \sqsubseteq D_0, \ldots, C_n \sqsubseteq D_n\}$ and $2^{Lit} = \{\mathsf{L}_0, \ldots, \mathsf{L}_m\}$, we choose a finite set of anonymous individuals $\mathsf{W} \subseteq \Delta \setminus (\mathsf{Ind} \cup \{c_u\})$

such that for each $C_i \sqsubseteq D_i \in \mathcal{G}$ and each $\mathsf{L}_j \in 2^{Lit}$ with $i = 0, \dots, n$ and $j = 0, \dots, m$ there exists a distinct individual $c_{i,j} \in \mathsf{W}$. The additional assertions are defined as follows

$$\mathcal{G}^{\neg} := \{(C \sqcap \neg D)(c) \mid C \sqsubseteq D \in \mathcal{G}, c \in \mathsf{Ind} \cup \{c_u\} \cup \mathsf{W}\}.$$

In the following lemma we show that we can rename an interpretation without changing its dynamic type w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ such that if a CI $C \sqsubseteq D$ is violated then a corresponding assertion $(C \sqcap \neg D)(c)$ from $\mathcal{G}^{\neg}$ is satisfied.

Lemma 63. *For every $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ there exists a $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ such that it holds that*

$$\mathsf{d\text{-}type}(\mathcal{I}) = \mathsf{d\text{-}type}(\mathcal{J})$$

for the dynamic types w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and for all $C \sqsubseteq D \in \mathcal{G}$ and all $\mathsf{L} \in 2^{Lit}$ it holds that if $(\neg(C \sqsubseteq D), \mathsf{L}) \in \mathsf{d\text{-}type}(\mathcal{I})$, then there exists $c \in \mathsf{Ind} \cup \mathsf{W}$ such that $\mathcal{J}^{\mathsf{L}} \models (C \sqcap \neg D)(c)$.

Proof. Let $\mathcal{I} \in \mathcal{M}(\mathcal{K})$. For each $C_i \sqsubseteq D_i \in \mathcal{G}$ and each $\mathsf{L}_j \in 2^{Lit}$ such that $(\neg(C_i \sqsubseteq D_i), \mathsf{L}_j) \in \mathsf{d\text{-}type}(\mathcal{I})$ we choose exactly one domain element $d_{i,j} \in \Delta$ with $d_{i,j} \in (C_i \sqcap \neg D_i)^{\mathcal{I}^{\mathsf{L}_j}}$. Let V be the set of all chosen $d_{i,j}$'s. It holds that $|V| \leq |\mathcal{G}| \cdot |2^{Lit}|$.

Let $\hat{\iota} : V \rightarrow \mathsf{Ind} \cup \{c_u\} \cup \mathsf{W}$ be a total injective function assigning to each element $d \in V$ an element $\hat{\iota}(d) \in \mathsf{Ind} \cup \{c_u\} \cup \mathsf{W}$ such that $\hat{\iota}(d) = d$ if $d \in \mathsf{Ind} \cup \{c_u\}$ and $\hat{\iota}(d) \notin \mathsf{Ind} \cup \{c_u\}$ if $d \notin \mathsf{Ind} \cup \{c_u\}$ for all $d \in V$. Now, let $\iota : \Delta \rightarrow \Delta$ be a bijection such that $\iota(a) = a$ for all $a \in \mathsf{Ind} \cup \{c_u\}$ and $\iota(d) = \hat{\iota}(d)$ for all $d \in V$. Clearly, such functions $\hat{\iota}$ and ι exist.

Let $\mathcal{J} := \iota(\mathcal{I})$ be the renaming of $\mathcal{I}$ w.r.t. ι according to Definition 19. Since $\iota(a) = a$ for all $a \in \mathsf{Ind} \cup \{c_u\}$, it is easy to see that Proposition 20 yields $\mathcal{J} \in \mathcal{M}(\mathcal{K})$ and $\mathsf{d\text{-}type}(\mathcal{I}) = \mathsf{d\text{-}type}(\mathcal{J})$.

Let $(\neg(C \sqsubseteq D), \mathsf{L}) \in \mathsf{d\text{-}type}(\mathcal{I}) = \mathsf{d\text{-}type}(\mathcal{J})$. There exists an element $d \in V$ such that $d \in (C \sqcap \neg D)^{\mathcal{I}^{\mathsf{L}}}$ and consequently $\iota(d) \in (C \sqcap \neg D)^{\mathcal{J}^{\mathsf{L}}}$. By construction we have $\iota(d) \in \mathsf{Ind} \cup \mathsf{W}$. Therefore, there exists an assertion of the form $(C \sqcap \neg D)(c) \in \mathcal{G}^{\neg}$ such that $\mathcal{J}^{\mathsf{L}} \models (C \sqcap \neg D)(c)$. □

A set $\mathfrak{d} \subseteq (\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^{\neg} \cup \neg\mathcal{G}^{\neg}) \times 2^{Lit}$ is called *admissible* if the following conditions are satisfied:

1. $(\varrho, \mathsf{L}) \in \mathfrak{d}$ iff $(\neg\varrho, \mathsf{L}) \notin \mathfrak{d}$ for all $\varrho \in \widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^{\neg} \cup \neg\mathcal{G}^{\neg}$ and all $\mathsf{L} \in 2^{Lit}$ (with $\neg\neg\varrho := \varrho$);
2. If $(\neg(C \sqsubseteq D), \mathsf{L}) \in \mathfrak{d}$ for some $C \sqsubseteq D \in \mathcal{G}$ and $\mathsf{L} \in 2^{Lit}$, then there exists $c \in \mathsf{Ind} \cup \{c_u\} \cup \mathsf{W}$ such that $((C \sqcap \neg D)(c), \mathsf{L}) \in \mathfrak{d}$.

With the second condition we achieve that the negated CIs can be omitted for the construction of $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$. Given $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ with $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and an admissible subset $\mathfrak{d}$ of $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^{\neg} \cup \neg\mathcal{G}^{\neg}) \times 2^{Lit}$ we are now ready to define $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$: For each concept name A and role name r occurring in $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and each $\mathsf{L} \in 2^{Lit}$ we introduce copies $A^{(\mathsf{L})}$ and $r^{(\mathsf{L})}$. For each subconcept C occurring in $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ and each $\mathsf{L} \in 2^{Lit}$ we have a new name $T^{(\mathsf{L})}_C$. The TBox $\mathcal{T}_{\text{sub}}$ contains a definition for all new names $T^{(\mathsf{L})}_C$ according to Figure 5 but where i is replaced by L and 0 by $\emptyset$. For ABox assertions and $\mathsf{L} \in 2^{Lit}$ we define the abbreviations $(C(a))^{(\mathsf{L})} := T^{(\mathsf{L})}_C(a)$, $(r(a,b))^{(\mathsf{L})} := r^{(\mathsf{L})}(a,b)$ and $(\neg r(a,b))^{(\mathsf{L})} := \neg r^{(\mathsf{L})}(a,b)$. The reduction TBox $\mathcal{T}^{\mathfrak{d}}_{\text{red}}$ is defined as follows:

$$\begin{aligned}
\mathcal{T}^{\mathfrak{d}}_{\text{red}} := {} & \mathcal{T}_{\text{sub}} \cup \\
& \{T^{(\emptyset)}_C \sqsubseteq T^{(\emptyset)}_D \mid C \sqsubseteq D \in \mathcal{T}\} \cup \\
& \{T^{(\mathsf{L})}_C \sqsubseteq T^{(\mathsf{L})}_D \mid (C \sqsubseteq D, \mathsf{L}) \in \mathfrak{d}\} \cup \\
& \{N \equiv \bigsqcup_{a \in \mathsf{Ind} \cup \{c_u\} \cup \mathsf{W}} \quad \}
\end{aligned}$$

The reduction ABox consists of the following components: First we consider the initial ABox

$$\mathcal{A}_{\mathrm{ini}} := \{\varphi^{(\emptyset)} \mid \varphi \in \mathcal{A}\}.$$

For each $\mathsf{L} \in 2^{Lit}$ we have $\mathcal{A}^{(\mathsf{L})} := \{\gamma^{(\mathsf{L})} \mid \gamma \in \mathsf{L}\}$ and $\mathcal{A}^{(\mathsf{L})}_{\mathrm{min}}$ consists of the following assertions:

- for each $a \in \mathsf{Ind}\cup\{c_u\}\cup\mathsf{W}$ and each concept name A occurring in the input with $\neg A(a) \notin \mathsf{L}$: $(A^{(\emptyset)} \rightarrow A^{(\mathsf{L})})(a)$ and if $A(a) \notin \mathsf{L}$: $(\neg A^{(\emptyset)} \rightarrow \neg A^{(\mathsf{L})})(a)$.
- for each pair of named elements $a, b \in \mathsf{Ind}\cup\{c_u\}\cup\mathsf{W}$ and every role name r with $\neg r(a,b) \notin \mathsf{L}$: $(\exists r^{(\emptyset)}.\{b\} \rightarrow \exists r^{(\mathsf{L})}.\{b\})(a)$ and if $r(a,b) \notin \mathsf{L}$: $(\forall r^{(\emptyset)}.\neg\{b\} \rightarrow \forall r^{(\mathsf{L})}.\neg\{b\})(a)$.

The ABox $\mathcal{A}^{\mathfrak{d}}$ contains for each pair $(\varphi, \mathsf{L}) \in \mathfrak{d}$ where φ is an ABox assertion an assertion $\varphi^{(\mathsf{L})}$. Putting everything together we get

$$\mathcal{A}^{\mathfrak{d}}_{\mathrm{red}} := \mathcal{A}_{\mathrm{ini}} \cup \mathcal{A}^{\mathfrak{d}} \cup \mathcal{A}^{(\mathsf{L}_0)}_{\mathrm{min}} \cup \cdots \cup \mathcal{A}^{(\mathsf{L}_m)}_{\mathrm{min}} \cup \mathcal{A}^{(\mathsf{L}_0)} \cup \cdots \cup \mathcal{A}^{(\mathsf{L}_m)}$$

and $\mathcal{K}^{\mathfrak{d}}_{\mathrm{red}} = (\mathcal{T}^{\mathfrak{d}}_{\mathrm{red}}, \mathcal{A}^{\mathfrak{d}}_{\mathrm{red}})$.

Lemma 64. *Let $\mathfrak{d}$ be an admissible subset of $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^\neg \cup \neg\mathcal{G}^\neg) \times 2^{Lit}$. There exists $\mathcal{I} \in \mathcal{M}(\mathcal{K})$ such that $\mathfrak{d} \setminus (\mathcal{G}^\neg \cup \neg\mathcal{G}^\neg) \times 2^{Lit} = \mathsf{d\text{-}type}(\mathcal{I})$ iff $\mathcal{K}^{\mathfrak{d}}_{red}$ is consistent.*

Proof. The claim follows from Lemma 63 and from the construction of $\mathcal{K}^{\mathfrak{d}}_{\mathrm{red}}$. □

The set of all dynamic types $\mathfrak{D}$ w.r.t. $\widehat{\mathcal{C}}(\mathcal{P}, c_u)$ can be computed by enumerating all admissible subsets of $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^\neg \cup \neg\mathcal{G}^\neg) \times 2^{Lit}$, constructing the $\mathcal{ALCO}$-KB $\mathcal{K}^{\mathfrak{d}}_{\mathrm{red}}$ for each admissible set $\mathfrak{d}$ and then checking whether $\mathcal{K}^{\mathfrak{d}}_{\mathrm{red}}$ is consistent. The size of the set $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^\neg \cup \neg\mathcal{G}^\neg) \times 2^{Lit}$ is exponential in the size of $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$. Thus, there are at most 2-exponentially many dynamic types and at most 3-exponentially many states in the abstract transition system $\widehat{T_{\mathcal{P}}}$. To decide whether a $\mathcal{ALCOK}$-LTL formula Φ is satisfiable in $\mathcal{P}$ we need to check whether there is an infinite path in $\widehat{T_{\mathcal{P}}}$ starting in an initial state such that the corresponding LTL-structure satisfies $\widehat{\Phi}$. This can be done using standard automata-based LTL model checking techniques.

Theorem 65. *Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$ be a restricted $\mathcal{ALCOK}$-Golog program and Φ an $\mathcal{ALCOK}$-LTL formula. The problem whether Φ is satisfiable in $\mathcal{P}$ or not is decidable in* 2ExpSpace.

Proof. Let $\mathcal{P} = (\mathcal{K}, \Sigma, \delta)$. We assume w.l.o.g. that all axioms in Φ are contained in $\mathcal{C}_{\mathsf{K}}(\mathcal{P})$.

We first sketch the main steps of the decision procedure. We construct $\widehat{T_{\mathcal{P}}} = (\widehat{Q}, \twoheadrightarrow, \widehat{I})$ and the propositional abstraction $\widehat{\Phi}$ of Φ. We need to deal with infinite words over the alphabet consisting of all subsets of $\mathsf{AP} = \{p_\varrho \mid \varrho \in \mathcal{C}_{\mathsf{K}}(\mathcal{P})\}$. Let $\mathcal{B}$ be a *(generalized) Büchi automaton* over the alphabet 2^{AP}. The language accepted by $\mathcal{B}$ is define as follows:

$$\mathcal{L}(\mathcal{B}) := \{\mathfrak{w} \in \left(2^{\mathsf{AP}}\right)^\omega \mid \mathcal{B} \text{ has an accepting run on } \mathfrak{w}\}.$$

Likewise for a given propositional LTL-formula ϕ over AP the language $\mathcal{L}(\phi) \subseteq \left(2^{\mathsf{AP}}\right)^\omega$ denotes the set of all LTL-structures $\mathfrak{L}$ such that $\mathfrak{L}, 0 \models \phi$. For the formula $\widehat{\Phi}$ a Büchi automaton $\mathcal{B}_{\widehat{\Phi}}$ can be constructed such that $\mathcal{L}(\widehat{\Phi}) = \mathcal{L}(\mathcal{B}_{\widehat{\Phi}})$. Now consider the following language

$$\mathcal{L}(\widehat{T_{\mathcal{P}}}) := \{\mathfrak{L}(\hat{\pi}) \mid \hat{\pi} = \hat{s}_0\hat{s}_1\hat{s}_2\cdots \text{ is an infinite path in } \widehat{T_{\mathcal{P}}}, \hat{s}_0 \in \widehat{I}\}.$$

We can show the following claim.

Claim I. It holds that $\mathcal{L}(\widehat{T_{\mathcal{P}}}) \cap \mathcal{L}(\mathcal{B}_{\widehat{\Phi}}) \neq \emptyset$ iff Φ is satisfiable in $\mathcal{P}$.

Proof. Assume $\mathcal{L}(\widehat{T}_{\mathcal{P}}) \cap \mathcal{L}(\mathcal{B}_{\widehat{\Phi}}) \neq \emptyset$. There exists $\mathfrak{w} \in \mathcal{L}(\widehat{T}_{\mathcal{P}}) \cap \mathcal{L}(\mathcal{B}_{\widehat{\Phi}})$ and an infinite path $\hat{\pi} = \hat{s}_0\hat{s}_1\hat{s}_2\cdots$ in $\widehat{T}_{\mathcal{P}}$ such that $\mathfrak{L}(\hat{\pi}) = \mathfrak{w}$ and $\mathfrak{L}(\hat{\pi}), 0 \models \widehat{\Phi}$ (since $\mathcal{L}(\widehat{\Phi}) = \mathcal{L}(\mathcal{B}_{\widehat{\Phi}})$). By Lemma 61 there exists a run π in $T_{\mathcal{P}}$ such that $\pi \bowtie \hat{\pi}$. From Lemma 62 and $\mathfrak{L}(\hat{\pi}), 0 \models \widehat{\Phi}$ it follows that $\mathfrak{I}(\pi), 0 \models \Phi$. Therefore Φ is satisfiable in $T_{\mathcal{P}}$.

Assume Φ is satisfiable in $\mathcal{P}$. There exists a run π in $T_{\mathcal{P}}$ such that $\mathfrak{I}(\pi), 0 \models \Phi$. By Lemma 61 there exists an infinite path $\hat{\pi}$ in $\widehat{T}_{\mathcal{P}}$ starting in an initial state such that $\pi \bowtie \hat{\pi}$. With Lemma 62 $\mathfrak{I}(\pi), 0 \models \Phi$ we get $\mathfrak{L}(\hat{\pi}), 0 \models \widehat{\Phi}$. Therefore $\mathfrak{L}(\hat{\pi}) \in \mathcal{L}(\widehat{T}_{\mathcal{P}}) \cap \mathcal{L}(\mathcal{B}_{\widehat{\Phi}})$. □

Checking whether $\mathcal{L}(\widehat{T}_{\mathcal{P}}) \cap \mathcal{L}(\mathcal{B}_{\widehat{\Phi}})$ is empty can be done by performing an emptiness test on the product Büchi automaton $\widehat{T}_{\mathcal{P}} \times \mathcal{B}_{\widehat{\Phi}}$. We show that a non-deterministic algorithm for this test exists that uses double exponential space in the size of $\mathcal{P}$ and Φ.

First we compute the set $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^{\neg} \cup \neg\mathcal{G}^{\neg}) \times 2^{Lit}$ that can be done in exponential time and space in the size of $\mathcal{P}$ (by Lemma 52). All 2-exponentially many admissible subsets of $(\widehat{\mathcal{C}}(\mathcal{P}, c_u) \cup \mathcal{G}^{\neg} \cup \neg\mathcal{G}^{\neg}) \times 2^{Lit}$ can be enumerated in double exponential space. For each admissible set $\mathfrak{d}$ the KB $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ can be constructed in polynomial time and space in the size of $\mathfrak{d}$ and $\mathcal{K}$. Since the size of $\mathfrak{d}$ is exponential in the size of the input, the size of $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ is also exponential in the size of the input. Consistency of $\mathcal{ALCO}$-KBs is in ExpTime. Therefore the consistency checks of the KBs $\mathcal{K}^{\mathfrak{d}}_{\text{red}}$ can be done in 2ExpTime and thus in 2ExpSpace. Consequently, the set of all dynamic types $\mathfrak{D}$ can be computed and stored in double exponential time and space, respectively. From Lemma 39 we know that there are at most polynomially many reachable subprograms of δ in the size of δ. Thus also $\mathsf{subg}(\delta)$ contains only polynomially many elements. For a state $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle \in \widehat{Q}$ in the abstract transition system it holds that $\mathfrak{t} \in \mathfrak{D} \times 2^{Lit}$, $\mathfrak{T} \subseteq \mathfrak{D} \times 2^{Lit}$ and $\rho \in \mathsf{subg}(\delta)$. It follows that

$$|\widehat{Q}| \leq |\mathfrak{D} \times 2^{Lit}| \cdot 2^{|\mathfrak{D} \times 2^{Lit}|} \cdot |\mathsf{subg}(\delta)|,$$

i.e. there are at most 3-exponentially many states in the size of $\mathcal{P}$ in $\widehat{T}_{\mathcal{P}}$. Given a state of the form $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$ the labeling $L_{\mathsf{AP}}(\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle)$ can be computed in linear time and space in the size of $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$. Given $\mathfrak{a} \in \mathsf{head}(\rho)$, executability and the successor state can be computed also in linear time and space in the size of $\langle(\mathfrak{t}, \mathfrak{T}), \rho\rangle$.

Let S be the set of states of $\mathcal{B}_{\widehat{\Phi}}$. $\mathcal{B}_{\widehat{\Phi}}$ can be constructed such that $|S| \leq 2^{|\widehat{\Phi}|} \cdot |\widehat{\Phi}|$.

We now proceed in the standard way of checking emptiness of the language accepted by $\widehat{T}_{\mathcal{P}} \times \mathcal{B}_{\widehat{\Phi}}$: We guess an infinite periodic accepting path in the product of the form

$$u_0 \ldots u_{n-1}(u_n \ldots u_{n+m})^{\omega}$$

such that $n \leq |\widehat{Q}| \cdot |S|$ and $m \leq |\widehat{Q}| \cdot |S|$.

First, we guess the two numbers n and m encoded in binary. They can be stored in 2-exponential space. Then we guess step by step the infinite path until the bounds n and m are reached where we do not need to keep the whole path in memory but only the current state and the previous one. As mentioned above the labeling of a state in $\widehat{T}_{\mathcal{P}}$ and the representation of a successor state can be computed in 2ExpSpace. This way we obtain a non-deterministic algorithm that uses at most 2-exponential space. □

An ExpSpace lower bound for the verification problem can be obtained by reducing the plan existence problem with conditional propositional actions which is ExpSpace-hard [Rin04]. The exact complexity of the problem remains open.

5 Related Work

So far, little work has been done on decidable verification of knowledge-based programs. De Giacomo, Lespérance and Patrizi [DLP13] present a class of epistemic Situation Calculus action theories for which they show decidability of μ-calculus properties, however they do not consider Golog and rely on a purely semantical definition of this class. On the propositional level, Lang and Zanuttini [LZ12] have investigated the complexity of verifying post-conditions of knowledge-based programs.

Finally, alternative approaches for reasoning about actions and programs using DLs were proposed. The formalization presented in [CDGLR11] for example adopts Levesque's functional view on knowledge bases, where all interactions with the agent's KB to happen through the two operations ASK (test evaluation) and TELL (update after action execution). While this allows for tractable solutions to the executability and projection problems for certain light-weight DLs, this non-declarative representation makes no distinction between world-changing and sensing actions as we do. Also, verification of temporal properties is not considered.

6 Conclusion

In this report, we introduced an action language for both physical and sensing actions based on the epistemic DL $\mathcal{ALCOK}$. We showed that under suitable restrictions, verifying LTL properties over possibly epistemic $\mathcal{ALCOK}$-axioms of knowledge-based Golog programs based on our action language is decidable. The main idea to obtain decidability is to syntactically limit the domain of the guarded pick operator to contain named objects only. Intuitively, under this restriction the agent won't be able to directly cope with unknown individuals. As seen in our running example, the agent is able to recognize whether or not there is *some unknown* fault, but there is no possibility to directly access it. We also showed that omitting any of the restrictions on the guarded pick operator leads to undecidability.

Furthermore, we investigated the complexity of projection as the basic reasoning task for executing knowledge-based programs. As future work, among other things, we want to investigate whether the obtained upper complexity bound of the verification problem can be improved, or it is actually tight.

References

[BCM$^+$03] Baader, Franz (Hrsg.) ; Calvanese, Diego (Hrsg.) ; McGuinness, Deborah L. (Hrsg.) ; Nardi, Daniele (Hrsg.) ; Patel-Schneider, Peter F. (Hrsg.): *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003

[BLM$^+$05] Baader, Franz ; Lutz, Carsten ; Miličić, Maja ; Sattler, Ulrike ; Wolter, Frank: Integrating Description Logics and Action Formalisms: First Results. In: Veloso, Manuela M. (Hrsg.) ; Kambhampati, Subbarao (Hrsg.): *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI 2005)*, AAAI Press, 2005, S. 572–577

[BZ13a] Baader, Franz ; Zarriess, Benjamin: Verification of Golog Programs over Description Logic Actions. In: Fontaine, Pascal (Hrsg.) ; Ringeissen, Christophe (Hrsg.) ; Schmidt, Renate A. (Hrsg.): *Proceedings of the Ninth International*

Symposium on Frontiers of Combining Systems (FroCoS'13) Bd. 8152, Springer-Verlag, 2013 (Lecture Notes in Artificial Intelligence)

[BZ13b] Baader, Franz ; Zarriess, Benjamin: Verification of Golog Programs over Description Logic Actions / Chair of Automata Theory, TU Dresden. Dresden, Germany, 2013 (13-08). – LTCS-Report. – See http://lat.inf.tu-dresden.de/research/reports.html.

[CDGLR11] Calvanese, Diego ; De Giacomo, Giuseppe ; Lenzerini, Maurizio ; Rosati, Riccardo: Actions and Programs over Description Logic Knowledge Bases: A Functional Approach. In: Lakemeyer, Gerhard (Hrsg.) ; McIlraith, Sheila A. (Hrsg.): *Knowing, Reasoning, and Acting: Essays in Honour of Hector J. Levesque*. College Publications, 2011

[CL06] Classen, Jens ; Lakemeyer, Gerhard: Foundations for Knowledge-Based Programs using ES. In: Doherty, Patrick (Hrsg.) ; Mylopoulos, John (Hrsg.) ; Welty, Christopher A. (Hrsg.): *Proceedings of the 10th Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, AAAI Press, 2006, 318–328

[CL08] Classen, Jens ; Lakemeyer, Gerhard: A Logic for Non-Terminating Golog Programs. In: Brewka, Gerhard (Hrsg.) ; Lang, Jérôme (Hrsg.): *Proceedings of the Eleventh International Conference on the Principles of Knowledge Representation and Reasoning (KR 2008)*, AAAI Press, 2008, S. 589–599

[Cla14] Classen, Jens: *Planning and Verification in the Agent Language Golog*, RWTH Aachen University, Diss., 2014

[DLL00] De Giacomo, Giuseppe ; Lespérance, Yves ; Levesque, Hector J.: ConGolog, a concurrent programming language based on the situation calculus. In: *Artificial Intelligence* 121 (2000), Nr. 1–2, S. 109–169

[DLN$^+$92] Donini, Francesco M. ; Lenzerini, Maurizio ; Nardi, Daniele ; Schaerf, Andrea ; Nutt, Werner: Adding Epistemic Operators to Concept Languages. In: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, October 25-29, 1992.*, 1992, S. 342–353

[DLN$^+$98] Donini, Francesco M. ; Lenzerini, Maurizio ; Nardi, Daniele ; Nutt, Werner ; Schaerf, Andrea: An Epistemic Operator for Description Logics. In: *Artif. Intell.* 100 (1998), Nr. 1-2, 225–274. `http://dx.doi.org/10.1016/S0004-3702(98)00009-5`. – DOI 10.1016/S0004–3702(98)00009–5

[DLP10] De Giacomo, Giuseppe ; Lespérance, Yves ; Pearce, Adrian R.: Situation Calculus Based Programs for Representing and Reasoning about Game Structures. In: Lin, Fangzhen (Hrsg.) ; Sattler, Ulrike (Hrsg.) ; Truszczynski, Miroslaw (Hrsg.): *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, AAAI Press, 2010, S. 445–455

[DLP12] De Giacomo, Giuseppe ; Lespérance, Yves ; Patrizi, Fabio: Bounded Situation Calculus Action Theories and Decidable Verification. In: Brewka, Gerhard (Hrsg.) ; Eiter, Thomas (Hrsg.) ; McIlraith, Sheila A. (Hrsg.): *Proceedings of the Thirtenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*, AAAI Press, 2012

[DLP13] De Giacomo, Giuseppe ; Lespérance, Yves ; Patrizi, Fabio: Bounded Epistemic Situation Calculus Theories. In: Rossi, Francesca (Hrsg.): *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, IJCAI/AAAI, 2013. – ISBN 978–1–57735–633–2

[FMC$^+$05] F. Baader ; M. Milicic ; C. Lutz ; U. Sattler ; F. Wolter: Integrating Description Logics and Action Formalisms for Reasoning about Web Services / Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology. Germany, 2005 (LTCS-05-02). – LTCS-Report. – See http://lat.inf.tu-dresden.de/research/reports.html.

[Lev84] Levesque, Hector J.: Foundations of a Functional Approach to Knowledge Representation. In: *Artif. Intell.* 23 (1984), Nr. 2, 155–212. `http://dx.doi.org/10.1016/0004-3702(84)90009-2`. – DOI 10.1016/0004–3702(84)90009–2

[Lev90] Levesque, Hector J.: All I Know: A Study in Autoepistemic Logic. In: *Artif. Intell.* 42 (1990), Nr. 2-3, 263–309. `http://dx.doi.org/10.1016/0004-3702(90)90056-6`. – DOI 10.1016/0004–3702(90)90056–6

[LL01] Levesque, Hector J. ; Lakemeyer, Gerhard: *The Logic of Knowledge Bases.* MIT Press, 2001

[LL04] Lakemeyer, Gerhard ; Levesque, Hector J.: Situations, Si! Situation Terms, No! In: Dubois, Didier (Hrsg.) ; Welty, Christopher A. (Hrsg.) ; Williams, Mary-Anne (Hrsg.): *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2004)*, AAAI Press, 2004, S. 516–526

[LL10] Lakemeyer, Gerhard ; Levesque, Hector J.: A semantic characterization of a useful fragment of the situation calculus with knowledge. In: *Artificial Intelligence* 175 (2010), Nr. 1, S. 142–164

[LRL$^+$97] Levesque, Hector J. ; Reiter, Raymond ; Lespérance, Yves ; Lin, Fangzhen ; Scherl, Richard B.: GOLOG: A Logic Programming Language for Dynamic Domains. In: *Journal of Logic Programming* 31 (1997), Nr. 1–3, S. 59–83

[LZ12] Lang, Jérôme ; Zanuttini, Bruno: Knowledge-Based Programs as Plans - The Complexity of Plan Verification. In: *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, 2012, 504–509

[Meh14] Mehdi, Anees ul: *Epistemic Reasoning in OWL 2 DL*, Karlsruher Instituts f́ur Technologie (KIT), Dissertation, 2014. – http://nbn-resolving.org/urn:nbn:de:swb:90-399632

[MH69] McCarthy, John ; Hayes, Patrick: Some philosophical problems from the standpoint of artificial intelligence. In: Meltzer, B. (Hrsg.) ; Michie, D. (Hrsg.): *Machine Intelligence 4*. New York : American Elsevier, 1969, S. 463–502

[Min67] Minsky, Marvin L.: *Computation: Finite and Infinite Machines.* Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1967. – ISBN 0–13–165563–9

[MR11] Mehdi, Anees ; Rudolph, Sebastian: Revisiting Semantics for Epistemic Extensions of Description Logics. In: Burgard, Wolfram (Hrsg.) ; Roth, Dan (Hrsg.): *AAAI*, AAAI Press, 2011

[Rei91] Reiter, Raymond: The Frame Problem in the Situation Calculus: A simple Solution (sometimes) and a Completeness Result for Goal Regression. In: *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy* (1991), S. 359–380

[Rei01a] Reiter, Raymond: *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* MIT Press, 2001

[Rei01b] Reiter, Raymond: On knowledge-based programming with sensing in the situation calculus. In: *ACM Trans. Comput. Log.* 2 (2001), Nr. 4, 433–457. `http://dx.doi.org/10.1145/502166.502167`. – DOI 10.1145/502166.502167

[Rin04] Rintanen, Jussi: Complexity of Planning with Partial Observability. In: *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), June 3-7 2004, Whistler, British Columbia, Canada*, 2004, 345–354

[Ros07] Rosati, Riccardo: The Limits of Querying Ontologies. In: *Database Theory - ICDT 2007, 11th International Conference, Barcelona, Spain, January 10-12, 2007, Proceedings*, 2007, 164–178

[SL03] Scherl, Richard B. ; Levesque, Hector J.: Knowledge, action, and the frame problem. In: *Artif. Intell.* 144 (2003), Nr. 1-2, 1–39. `http://dx.doi.org/10.1016/S0004-3702(02)00365-X`. – DOI 10.1016/S0004–3702(02)00365–X

[SLG14] Steigmiller, Andreas ; Liebig, Thorsten ; Glimm, Birte: Konclude: System Description. In: *Journal of Web Semantics (JWS)* (2014)

[TH06] Tsarkov, Dmitry ; Horrocks, Ian: FaCT++ Description Logic Reasoner: System Description. In: *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)* Bd. 4130, Springer, 2006 (Lecture Notes in Artificial Intelligence), 292–297

[ZC14] Zarriess, Benjamin ; Classen, Jens: Verifying CTL* Properties of Golog Programs over Local-effect Actions. In: Geffner, Hector (Hrsg.) ; Schaub, Torsten (Hrsg.) ; Friedrich, Gerhard (Hrsg.) ; O'Sullivan, Barry (Hrsg.): *Proceedings of the Twenty-First European Conference on Artificial Intelligence (ECAI 2014)*, IOS Press, 2014, S. 939–944

[ZC15] Zarriess, Benjamin ; Classen, Jens: Verification of Knowledge-Based Programs over Description Logic Actions. In: *Proceedings of IJCAI-15*, 2015