TECHNISCHE
UNIVERSITÄT
DRESDEN

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

LTCS–Report

Decidable Verification of Golog Programs over Non-Local Effect Actions

Extended Version

Benjamin Zarrieß Jens Claßen

LTCS-Report 15-19

This is an extended version of a paper in the Proceedings of AAAI-16.

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

Decidable Verification of Golog Programs over Non-Local Effect Actions [*]

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

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

December 14, 2015

Abstract

The Golog action programming language is a powerful means to express high-level behaviours in terms of programs over actions defined in a Situation Calculus theory. In particular for physical systems, verifying that the program satisfies certain desired temporal properties is often crucial, but undecidable in general, the latter being due to the language's high expressiveness in terms of first-order quantification and program constructs. So far, approaches to achieve decidability involved restrictions where action effects either had to be *context-free* (i.e. not depend on the current state), *local* (i.e. only affect objects mentioned in the action's parameters), or at least *bounded* (i.e. only affect a finite number of objects). In this paper, we present a new, more general class of action theories (called *acyclic*) that allows for context-sensitive, non-local, unbounded effects, i.e. actions that may affect an unbounded number of possibly unnamed objects in a state-dependent fashion. We contribute to the further exploration of the boundary between decidability and undecidability for Golog, showing that for acyclic theories in the two-variable fragment of first-order logic, verification of CTL* properties of programs over ground actions is decidable.

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

Contents

1 Introduction

When it comes to the design and programming of an autonomous agent, the Golog [LRL$^+$97] family of action languages offers a powerful means to express high-level behaviours in terms of complex programs whose basic building blocks are the primitive actions described in a Situation Calculus [Rei01] action theory. Golog's biggest advantage perhaps is the fact that a programmer can freely combine imperative control structures with non-deterministic constructs, leaving it to the system to resolve non-determinism in a suitable manner.

In particular when Golog is used to control physical robots, it is often crucial to verify a program against some specification of desired behaviour, for example in order to ensure liveness and safety properties, typically expressed by means of temporal formulas. Unfortunately, the general verification problem for Golog is undecidable due to the language's high expressivity in terms of first-order quantification, range of action effects, and program constructs. For this reason, there have recently been endeavours to identify restricted, but non-trivial fragments of Golog where verification (and hence other reasoning tasks such as projection) becomes decidable, while a great deal of expressiveness is retained.

So far, approaches to decidability [CLLZ14, ZC14, DLP12] required action theories to be restricted such that action effects are either *context-free* (not depend on the current state), *local* (only affect objects mentioned in the action's parameters), or at least *bounded* (only affect a finite number of objects). Examples that do *not* fall into either of these categories are the classical briefcase domain [Ped88] and exploding a bomb [LR97]: When a briefcase is moved, (unboundedly many, unmentioned) objects that are currently in it are being moved along, and if a bomb explodes, everything in its vicinity is destroyed.

In this paper, we extend the results from [ZC14] and present two new, more general classes of action theories over the decidable FOL fragment C^2 that also allow for context-sensitive, non-local, unbounded effects, i.e. actions that may affect an unbounded number of possibly unnamed objects in a state-dependent fashion. In our classes of action theories we do not impose any bound on the number of affected objects, but restrict the dependencies between fluents in the successor state axioms. This allows for a much wider range of application domains, including the above mentioned briefcase and bomb examples.

In a transportation domain such as the briefcase example, the action of moving a briefcase changes the location of objects represented by the fluent predicate *At*. To describe the actual set of objects affected one also has to refer to the fluent predicate *In* relating the briefcase to its content. Thus, the effect of the move action on *At* depends on *In*. The class of *acyclic theories* is obtained by disallowing cyclic dependencies between fluents, and another class we call *flat theories* is obtained by resorting to quantifier-free formulas for defining the set of affected objects. Both are syntactic restrictions and are decidable to check.

After proving that verification of CTL* properties is generally undecidable for Golog, even when restricted to ground actions and C^2, we then show that for our new classes of action theories, decidability can be achieved. The proof introduces a new, compact form of regression of formulas and establishes an abstraction to propositional model checking.

2 Preliminaries

2.1 Basic action theories in $\mathcal{ES}$ based on C^2

In this subsection we recall the main definitions of a fragment of the first-order modal logic $\mathcal{ES}$ [LL04, LL10] for reasoning about actions. We consider Situation Calculus *Basic Action Theories* (BATs) [Rei01] formulated in $\mathcal{ES}$ where the base logic is restricted to the *two variable fragment with equality and counting* of FOL named C^2.

We start by defining a set of *terms*.

Definition 1 (terms)**.** In our language we consider 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 *object constant symbols* (i.e. 0-ary function symbols);
- a countably infinite set N_A of *action function symbols* with arguments of sort object;

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 $\mathcal{N}_O$, and those of sort action by $\mathcal{N}_A$. ▲

To build formulas we consider *fluent* predicate symbols with at most two arguments of sort object. Fluents vary as the result of actions. Formulas are then built using the usual logical connectives and in addition we have two modal operators $[\cdot]$ and $\Box$ for referring to future situations, where $\Box\phi$ says that ϕ holds after any sequence of actions, and $[t]\phi$ means that ϕ holds after executing action t.

Definition 2 (formulas)**.** Let N_F be a set of fluent predicate symbols. The set of formulas is defined as the least set satisfying the following conditions:

- If $t_1, ..., t_k$ are terms and $F \in N_F$ a k-ary predicate symbol with $0 \leq k \leq 2$, then $F(t_1, ..., t_k)$ is a formula.

- If t_1 and t_2 are terms, then $t_1 = t_2$ is a formula.
- If ϕ_1 and ϕ_2 are formulas, x a variable and t a term of sort action, then
  - $\phi_1 \wedge \phi_2$, $\neg\phi_1$, $\forall x.\phi_1$, $\exists^{\leq m} x.\phi_1$ and $\exists^{\geq m} x.\phi_1$ with $m \in \mathbb{N}$ are formulas and
  - $\Box\phi_1$ (ϕ_1 always holds) and $[t]\phi_1$ (ϕ_1 holds after executing t) are formulas.

We understand $\vee$, $\exists$, $\exists^{=m}$, $\supset$, $\equiv$ and *true* and *false* as the usual abbreviations. A formula is called *fluent formula* if it contains no $\Box$ and no $[\cdot]$. A *fluent sentence* is a fluent formula without free variables. A C^2*-fluent formula* is a fluent formula that contains no terms of sort action and at most two variables. We assume that in a C^2-fluent formula only the variable symbols x and y are allowed to occur. ▲

The semantics of formulas is defined in terms of *worlds*.

Definition 3 (world)**.** Let $\mathcal{P}_F$ be the set of all primitive formulas $F(n_1, ..., n_k)$, where F is a k-ary fluent with $0 \leq k \leq 2$ and the n_i are standard names of sort object. Let $\mathcal{Z} := \mathcal{N}_A^*$. A *world* w is a mapping of the form

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

The set of all worlds is denoted by $\mathcal{W}$. ▲

A world thus maps primitive formulas to truth values.

We use the symbol $\langle\rangle$ to denote the empty sequence of action standard names. We are now equipped to define the truth of formulas:

Definition 4 (truth of formulas)**.** Given a world $w \in \mathcal{W}$ and a closed formula ψ, we define $w \models \psi$ as $w, \langle\rangle \models \psi$, where for any $z \in \mathcal{Z}$:

1. $w, z \models F(n_1, \ldots, n_k)$ iff $w[F(n_1, \ldots, n_k), z] = 1$;
2. $w, z \models (n_1 = n_2)$ iff n_1 and n_2 are identical;
3. $w, z \models \psi_1 \wedge \psi_2$ iff $w, z \models \psi_1$ and $w, z \models \psi_2$;
4. $w, z \models \neg\psi$ iff $w, z \not\models \psi$;
5. $w, z \models \forall x.\phi$ iff $w, z \models \phi^x_n$ for all $n \in \mathcal{N}_x$;
6. $w, z \models \exists^{\leq m} x.\phi$ iff $|\{n \in \mathcal{N}_x \mid w, z \models \phi^x_n\}| \leq m$;
7. $w, z \models \exists^{\geq m} x.\phi$ iff $|\{n \in \mathcal{N}_x \mid w, z \models \phi^x_n\}| \geq m$;
8. $w, z \models \Box\psi$ iff $w, z \cdot z' \models \psi$ for all $z' \in \mathcal{Z}$;
9. $w, z \models [t]\psi$ iff $w, z \cdot t \models \psi$;

▲

Above, $\mathcal{N}_x$ refers to the set of all standard names of the same sort as x. We moreover use ϕ^x_n to denote the result of simultaneously replacing all free occurrences of x in ϕ by n. Note that by rule 2 above, the unique names assumption (UNA) for actions and object constants is part of our semantics. In the following we use the notation $\vec{x}$ and $\vec{y}$ for sequences of object variables and $\vec{v}$ for a sequence of object terms.

In the following we will often omit leading universal quantifiers and parentheses. We assume the following precedence order of the logical connectives and quantifiers: $[\cdot], \neg, \wedge, \vee, \forall, \exists, \supset, \equiv, \square$, i.e. $[\cdot]$ has the highest priority and $\square$ the lowest.

We now define a basic action theory as a set of axioms of a pre-defined structure in order to model a dynamic application domain.

Definition 5. A C^2-*basic action theory* (C^2-BAT) $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}$ describes the dynamics of a specific application domain, where

1. $\mathcal{D}_0$, *the initial theory*, is a finite set of C^2-fluent sentences describing the initial state of the world.

2. $\mathcal{D}_{\text{post}}$ is a finite set of *successor state axioms* (SSAs), one for each fluent relevant to the application domain, incorporating Reiter's [Rei01] 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

$$\forall a.\forall \vec{x}.\square\Big(\big([a]F(\vec{x})\big) \equiv \gamma_F^+ \vee \big(F(\vec{x}) \wedge \neg\gamma_F^-\big)\Big)$$

   where the *positive effect condition* γ_F^+ and *negative effect condition* γ_F^- are fluent formulas. We require that γ_F^+ and γ_F^- are (possibly empty) disjunctions of formulas of the form $\exists\vec{y}.\big(a = A(\vec{v}) \wedge \phi \wedge \phi'\big)$ such that

   (a) $\exists\vec{y}.\big(a = A(\vec{v}) \wedge \phi \wedge \phi'\big)$ contains the free variables $\vec{x}$ and a and no other free variables;

   (b) $A(\vec{v})$ is an action term and $\vec{v}$ contains $\vec{y}$;

   (c) ϕ is a fluent formula with no terms of sort action and the number of variable symbols in ϕ that do not occur in $\vec{v}$ or occur bounded in ϕ is less or equal two;

   (d) ϕ' is a fluent formula with free variables among $\vec{v}$, no terms of sort action, and at most two bounded variables.

   The formula ϕ is called *effect descriptor* and ϕ' is called *context condition.*

▲

The restrictions 2a and 2b on SSAs are wlog and describe the usual syntactic form of SSAs. Intuitively, the effect descriptor ϕ possibly defines a complex set of objects (or a set of pairs of objects in case F is a binary fluent) that are added or deleted to or from the relational fuent F, respectively, if $A(\vec{v})$ is executed. Provided that free occurrences of variables in ϕ that occur as arguments of $A(\vec{v})$ are instantiated, the condition 2c ensures definability of the (instantiated) effect descriptor in our base logic C^2. In contrast to the effect descriptor the context condition ϕ' only tells us *whether* $A(\vec{v})$ has an effect on F but *not which* objects are actually affected. As for the effect descriptor the condition 2d ensures that after instantiation of the action, the context condition is a sentence in C^2. Therefore the variables $\vec{x}$ mentioned in 2a may have free occurrences in ϕ but not in ϕ'.

Example 6. We consider a domain with *servers* hosting *virtual machines* and *processes* that might be classified as *malware.* There is a fluent $\mathit{Avail}(x)$ denoting processes x that are currently available, and $\mathit{Ovl}(x)$ for a server x that is overloaded. $\mathit{Hosts}(x, y)$ furthermore says that a server x hosts a virtual machine or a process y, and $\mathit{Runs}(x, y)$ is true for a virtual machine x running a process y.

The agent can migrate a virtual machine (v) hosted on server (s) to a server (s') if s' is not overloaded using the action $Migr(v, s, s')$. We also have exogenous actions, i.e. actions not under the control of the agent, of the form $Att(s)$, saying that a server is subject of an attack causing it to be overloaded, and $Repair(s)$, which returns the server s to its original state. Figure 2 exemplarily shows the effect conditions for the fluents $Avail(x)$, $Ovl(x)$ and $Hosts(x, y)$. The effect descriptors are underlined with a solid line and the context conditions with a dashed line. Consider the execution of $Migr(vm, s_1, s_2)$ in an initial situation incompletely described by the axioms in Figure 1. The action has an effect on the fluent $Avail(x)$ because the context condition is satisfied, i.e. the target server s_2 is not overloaded. The instantiated effect descriptor yields that for all objects d, $Avail(d)$ is *true after* doing the action if $Runs(vm, d)$ is *true before* doing the action. Thus, all processes running on vm become available. Furthermore, the fluent $Hosts(x, y)$ is also affected: all processes running on vm are now hosted by s_2 and no longer by s_1. A BAT based on these axioms for example entails

$$[Migr(vm, s_1, s_2)]\big(\forall x.Runs(vm, x) \supset Avail(x)\big).$$

▲

$$\begin{aligned}
&Hosts(s_1, vm), Hosts(s_1, p), Runs(vm, p), \neg Avail(p)\\
&Server(s_2), \neg Ovl(s_2), \forall y.\exists^{\leq 1}x.Hosts(x, y),\\
&\forall x, y.Hosts(x, y) \supset Server(x) \wedge \big(Proc(y) \vee VM(y)\big)
\end{aligned}$$

Figure 1: Example initial theory

$$\begin{aligned}
\gamma^{+}_{Avail} &:= \exists v, s, s'.\big(\, a = Migr(v, s, s') \wedge\\
&\qquad\qquad \underline{Runs(v, x)} \wedge \text{\dashuline{$\neg Ovl(s')$}}\,\big) \vee\\
&\qquad \exists s.\big(a = Repair(s) \wedge \underline{Hosts(s, x) \wedge Proc(x)}\big);\\
\gamma^{-}_{Avail} &:= \exists s.\big(\, a = Att(s) \wedge \underline{Hosts(s, x) \wedge Proc(x)} \wedge\\
&\qquad\qquad \text{\dashuline{$\exists y.Hosts(s, y) \wedge Malware(y)$}}\,\big);\\
\gamma^{+}_{Ovl} &:= \exists s.\big(\, a = Att(s) \wedge \underline{x = s} \wedge\\
&\qquad\qquad \text{\dashuline{$\exists y.Hosts(s, y) \wedge Malware(y)$}}\,\big);\\
\gamma^{-}_{Ovl} &:= \exists s.\big(a = Repair(s) \wedge \underline{x = s}\big);\\
\gamma^{+}_{Hosts} &:= \exists v, s, s'.\big(\, a = Migr(v, s, s') \wedge \underline{x = s' \wedge}\\
&\qquad\qquad \underline{\big(Runs(v, y) \vee y = v\big)} \wedge \text{\dashuline{$\neg Ovl(s')$}}\,\big);\\
\gamma^{-}_{Hosts} &:= \exists v, s, s'.\big(\, a = Migr(v, s, s') \wedge \underline{x = s \wedge}\\
&\qquad\qquad \underline{\big(Runs(v, y) \vee y = v\big)} \wedge \text{\dashuline{$\neg Ovl(s')$}}\,\big)
\end{aligned}$$

Figure 2: Example effect conditions

2.2 Golog programs and the verification problem

In a Golog program we combine atomic actions, whose effects are defined in a C^2-BAT, and tests using a set of programming constructs to define a complex action. Here we define program expressions as extra-logical expressions.

Definition 7 (Golog program)**.** A *program expression* δ is built according to the following grammar

$$\delta ::= \langle\rangle \mid t \mid \psi? \mid \delta;\delta \mid \delta|\delta \mid \delta^* \mid \delta\|\delta$$

A program expression can thus be the *empty program* $\langle\rangle$, a ground action term t, a *test* $\psi?$, where ψ is a C^2 fluent sentence, or constructed from subprograms be means of *sequence* $\delta;\delta$, *non-deterministic choice* $\delta|\delta$, *non-deterministic iteration* δ^* and *interleaving* $\delta\|\delta$.

A *Golog program* $\mathcal{G} = (\mathcal{D}, \delta)$ consists of a C^2-BAT $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}$ and a program expression δ where all fluents occurring in $\mathcal{D}$ and δ have an SSA in $\mathcal{D}_{\text{post}}$.

To handle termination and failure of a program we use two 0-ary fluents $Final$ and $Fail$ and two 0-ary action functions ϵ and $\mathfrak{f}$ and include the SSAs $\Box[a]Final \equiv a = \epsilon \vee Final$ and $\Box[a]Fail \equiv a = \mathfrak{f} \vee Fail$ in $\mathcal{D}_{\text{post}}$. Furthermore, we require that $\neg Final \in \mathcal{D}_0$ and $\neg Fail \in \mathcal{D}_0$, and that the fluents $Final$, $Fail$ and actions ϵ and $\mathfrak{f}$ do not occur in δ. ▲

Next, we define the semantics of programs following [CL08].

Definition 8 (program semantics)**.** A *configuration* $\langle z, \delta\rangle$ consists of an action sequence $z \in \mathcal{Z}$ and a program expression δ, where intuitively z is the history of actions that have already been performed, while δ is the program that remains to be executed. The *transition relation* $\xrightarrow{w}$ *among configurations*, given a world $w \in \mathcal{W}$, is defined by induction on the size of program expressions as the least set satisfying the following conditions:

1. $\langle z, t\rangle \xrightarrow{w} \langle z \cdot t, \langle\rangle\rangle$;
2. $\langle z, \delta_1; \delta_2\rangle \xrightarrow{w} \langle z \cdot t, \gamma; \delta_2\rangle$, if $\langle z, \delta_1\rangle \xrightarrow{w} \langle z \cdot t, \gamma\rangle$;
3. $\langle z, \delta_1; \delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$, if $\langle z, \delta_1\rangle \in \mathsf{Fin}(w)$ and $\langle z, \delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$;
4. $\langle z, \delta_1|\delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$, if $\langle z, \delta_1\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$ or $\langle z, \delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$;
5. $\langle z, \delta^*\rangle \xrightarrow{w} \langle z \cdot t, \gamma; \delta^*\rangle$, if $\langle z, \delta\rangle \xrightarrow{w} \langle z \cdot t, \gamma\rangle$.
6. $\langle z, \delta_1\|\delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\|\delta_2\rangle$, if $\langle z, \delta_1\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$;
7. $\langle z, \delta_1\|\delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta_1\|\delta'\rangle$, if $\langle z, \delta_2\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle$;

The set of final configurations $\mathsf{Fin}(w)$ w.r.t. a world w is the smallest set such that

1. $\langle z, \langle\rangle\rangle \in \mathsf{Fin}(w)$.
2. $\langle z, \psi?\rangle \in \mathsf{Fin}(w)$ if $w, z \models \psi$;
3. $\langle z, \delta_1; \delta_2\rangle \in \mathsf{Fin}(w)$ if $\langle z, \delta_1\rangle \in \mathsf{Fin}(w)$ and $\langle z, \delta_2\rangle \in \mathsf{Fin}(w)$;
4. $\langle z, \delta_1|\delta_2\rangle \in \mathsf{Fin}(w)$ if $\langle z, \delta_1\rangle \in \mathsf{Fin}(w)$ or $\langle z, \delta_2\rangle \in \mathsf{Fin}(w)$;
5. $\langle z, \delta^*\rangle \in \mathsf{Fin}(w)$;
6. $\langle z, \delta_1\|\delta_2\rangle \in \mathsf{Fin}(w)$ if $\langle z, \delta_1\rangle \in \mathsf{Fin}(w)$ and $\langle z, \delta_2\rangle \in \mathsf{Fin}(w)$;

The set of *failing configurations* w.r.t. a world w is given by

$$\mathsf{Fail}(w) := \{\langle z, \delta\rangle \mid \langle z, \delta\rangle \notin \mathsf{Fin}(w),\ \text{there is no } \langle z \cdot t, \delta'\rangle \text{ s.t. } \langle z, \delta\rangle \xrightarrow{w} \langle z \cdot t, \delta'\rangle\}.$$

We extend final and failing configurations with additional transitions by defining an extension of $\xrightarrow{w}$. Let $w \in \mathcal{W}$. The *extended transition relation* $\overset{w}{\hookrightarrow}$ *among configurations* is defined as the least set satisfying the following conditions

1. $\langle z, \delta \rangle \overset{w}{\hookrightarrow} \langle z \cdot t, \delta' \rangle$, if $\langle z, \delta \rangle \overset{w}{\to} \langle z \cdot t, \delta' \rangle$;

2. $\langle z, \delta \rangle \overset{w}{\hookrightarrow} \langle z \cdot \epsilon, \langle \rangle \rangle$, if $\langle z, \delta \rangle \in \mathsf{Fin}(w)$;

3. $\langle z, \delta \rangle \overset{w}{\hookrightarrow} \langle z \cdot \mathfrak{f}, \delta \rangle$, if $\langle z, \delta \rangle \in \mathsf{Fail}(w)$.

Let $\overset{w}{\hookrightarrow}^*$ denote the reflexive and transitive closure of $\overset{w}{\hookrightarrow}$.

The *set of reachable configurations* from $\langle \langle \rangle, \delta \rangle$ in w is given by

$$\mathsf{Reach}(w, \delta) := \{ \langle z, \delta' \rangle \mid \langle \langle \rangle, \delta \rangle \overset{w}{\hookrightarrow}^* \langle z, \delta' \rangle \}.$$

Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a Golog program and $w \in \mathcal{W}$ a world with $w \models \mathcal{D}$. Execution of δ in w yields the *transition system of* $\mathcal{G}$ *w.r.t.* w given by

$$\mathsf{T}_\delta^w = \big(\mathsf{Reach}(w, \delta), \overset{w,\delta}{\hookrightarrow}\big),$$

where $\overset{w,\delta}{\hookrightarrow}$ is the restriction of $\overset{w}{\hookrightarrow}$ to configurations in $\mathsf{Reach}(w, \delta)$. ▲

A *path* π in $\mathsf{T}_\delta^w = \big(\mathsf{Reach}(w, \delta), \overset{w,\delta}{\hookrightarrow}\big)$ starting in $\langle z_0, \rho_0 \rangle \in \mathsf{Reach}(w, \delta)$ is an infinite sequence of the form

$$\pi = \langle z_0, \rho_0 \rangle \overset{w,\delta}{\hookrightarrow} \langle z_1, \rho_1 \rangle \overset{w,\delta}{\hookrightarrow} \langle z_2, \rho_2 \rangle \overset{w,\delta}{\hookrightarrow} \cdots .$$

For π and $j \in \{0, 1, 2, \ldots\}$ we denote the path

$$\langle z_j, \rho_j \rangle \overset{w,\delta}{\hookrightarrow} \langle z_{j+1}, \rho_{j+1} \rangle \overset{w,\delta}{\hookrightarrow} \langle z_{j+2}, \rho_{j+2} \rangle \overset{w,\delta}{\hookrightarrow} \cdots$$

by $\pi[j..]$. The *set of all paths* starting in $\langle z, \rho \rangle$ is denoted by $\mathsf{Paths}(\langle z, \rho \rangle, \mathsf{T}_\delta^w)$.

We are now ready to formulate temporal properties of transition systems.

Definition 9 (temporal properties of programs)**.** We define *temporal formulas*, whose syntax is the same as for propositional CTL*, but in place of propositions we allow for C^2-fluent sentences:

$$\Phi ::= \psi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathsf{E}\Psi \tag{1}$$
$$\Psi ::= \Phi \mid \neg\Psi \mid \Psi \wedge \Psi \mid \mathsf{X}\Psi \mid \Psi \,\mathsf{U}\, \Psi \tag{2}$$

Above, ψ can be any C^2-fluent sentence. We call formulas according to (1) *temporal state formulas*, and formulas according to (2) *temporal path formulas.* We use the usual abbreviations $\mathsf{A}\Psi$ (Ψ holds on *all* paths) for $\neg\mathsf{E}\neg\Psi$, $\mathsf{F}\Psi$ (*eventually* Ψ holds) for $\top \,\mathsf{U}\, \Psi$ and $\mathsf{G}\Psi$ (*globally* Ψ) for $\neg\mathsf{F}\neg\Psi$.

Let Φ be a temporal state formula, $\mathsf{T}_\delta^w = \big(\mathsf{Reach}(w, \delta), \overset{w,\delta}{\hookrightarrow}\big)$ the transition system of a program $\mathcal{G} = (\mathcal{D}, \delta)$ w.r.t. a world w with $w \models \mathcal{D}$ and $\langle z, \rho \rangle \in \mathsf{Reach}(w, \delta)$.

Truth of Φ in $\mathsf{T}_\delta^w, \langle z, \rho \rangle$, denoted by $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \Phi$, is defined as follows:

- $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \psi$ iff $w, z \models \psi$;
- $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \neg\Phi$ iff $\mathsf{T}_\delta^w, \langle z, \rho \rangle \not\models \Phi$;
- $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \Phi_1 \wedge \Phi_2$ iff $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \Phi_1$ and $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \Phi_2$;
- $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \mathsf{E}\Psi$ iff there exists $\pi \in \mathsf{Paths}(\langle z, \rho \rangle, \mathsf{T}_\delta^w)$ such that $\mathsf{T}_\delta^w, \pi \models \Psi$.

Let Ψ be a temporal path formula, T_δ^w and $\langle z, \rho \rangle$ as above, and $\pi \in \mathsf{Paths}(\langle z, \rho \rangle, \mathsf{T}_\delta^w)$. Truth of Ψ in T_δ^w, π, denoted by $\mathsf{T}_\delta^w, \pi \models \Psi$, is defined as follows:

- $\mathsf{T}_\delta^w, \pi \models \Phi$ iff $\mathsf{T}_\delta^w, \langle z, \rho \rangle \models \Phi$;
- $\mathsf{T}_\delta^w, \pi \models \neg\Psi$ iff $\mathsf{T}_\delta^w, \pi \not\models \Psi$;
- $\mathsf{T}_\delta^w, \pi \models \Psi_1 \wedge \Psi_2$ iff $\mathsf{T}_\delta^w, \pi \models \Psi_1$ and $\mathsf{T}_\delta^w, \pi \models \Psi_2$;
- $\mathsf{T}_\delta^w, \pi \models \mathsf{X}\Psi$ iff $\mathsf{T}_\delta^w, \pi[1..] \models \Psi$;
- $\mathsf{T}_\delta^w, \pi \models \Psi_1 \,\mathsf{U}\, \Psi_2$ iff $\exists k \geq 0 : \mathsf{T}_\delta^w, \pi[k..] \models \Psi_2$ and $\forall j, 0 \leq j < k : \mathsf{T}_\delta^w, \pi[j..] \models \Psi_1$.

▲

The verification problem is defined as follows.

Definition 10 (verification problem)**.** Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a Golog program and Φ a temporal state formula. Φ is *valid* in $\mathcal{G}$ iff for all worlds $w \in \mathcal{W}$ with $w \models \mathcal{D}$ it holds that $\mathsf{T}_\delta^w, \langle\langle\rangle, \delta\rangle \models \Phi$. Φ is *satisfiable* in $\mathcal{G}$ iff there exists $w \in \mathcal{W}$ with $w \models \mathcal{D}$ such that $\mathsf{T}_\delta^w, \langle\langle\rangle, \delta\rangle \models \Phi$. ▲

Example 11. Consider the program expressions in Figure 3. In δ_{avail} the virtual machine vm is migrated from server s_1 to server s_2 if s_1 hosts vm and is overloaded and vice versa if s_2 is overloaded. δ_{exo} consists of the exogenous attack and repair actions. To describe the actions that occur in the domain, both parts δ_{avail} and δ_{exo} are concurrently executed in infinite loops. A temporal property one might want to verify for the Golog program consisting of the C^2-BAT described in Example 6 and the program expression δ_{domain} could be:

$$\begin{aligned}&\mathsf{E}\big(\mathsf{GF}\big(\mathit{Ovl}(s_1) \wedge \mathit{Ovl}(s_2)\big) \supset \\ &\mathsf{E}\big(\mathsf{GF}\big(\forall x.\mathit{Runs}(vm, x) \supset \mathit{Avail}(x)\big) \wedge \mathsf{GF}\big(\mathit{Ovl}(s_1) \wedge \mathit{Ovl}(s_2)\big)\big).\end{aligned}$$

Validity of this property ensures that if it is possible that both servers are both infinitely often available, then it is possible that in addition also all processes running on vm are infinitely often available. ▲

$$\begin{aligned}\delta_{avail} :=\ & \exists x.(\mathit{Hosts}(x, vm) \wedge \mathit{Ovl}(x))?;\\ & \big(\mathit{Hosts}(s_1, vm)?; \mathit{Migr}(vm, s_1, s_2) \mid \\ & \ \mathit{Hosts}(s_2, vm)?; \mathit{Migr}(vm, s_2, s_1)\big)\\ \delta_{exo} :=\ & \big(\mathit{Att}(s_1) \mid \mathit{Att}(s_2) \mid \mathit{Repair}(s_1) \mid \mathit{Repair}(s_2)\big)\\ \delta_{domain} :=\ & \big[\big(\delta_{avail}\big)^*; \bot?\big] \parallel \big[\big(\delta_{exo}\big)^*; \bot?\big]\end{aligned}$$

Figure 3: Example program

3 (Un-)decidability of Verification

3.1 Undecidability in the General Case

As shown in [GS07], the projection problem that asks for a sequence of ground actions over a C^2-BAT whether a given C^2-fluent sentence holds after executing that sequence, is decidable. Unfortunately, verification for programs over ground actions is not:

We show that the verification problem is undecidable using a reduction of the halting problem of two-counter machines.

Theorem 12. *The verification problem is undecidable.*

Proof. 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 $\ell \in \{0, 1\}$. There are three types of instructions:

- $\mathsf{Inc}(\ell, i)$: Increment c_ℓ by one and jump to instruction J_i.
- $\mathsf{Dec}(\ell, i, j)$: If $c_\ell = 0$ jump to J_i, else if $c_\ell > 0$ decrement c_ℓ 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 that 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].

We define a Golog program simulating M using the following signature

- two unary fluents C_0 and C_1 one for each counter;
- a 0-ary fluent Halt, and 0-ary fluents $J_0, \ldots, J_m$ one for each instruction
- a binary rigid predicate Adj and a constant $\mathbf{0} \in \mathcal{N}_O$.

To represent the values of the counters in a world we define an infinite chain of objects starting in $\mathbf{0}$ using the binary predicate Adj. We ensure that in each situation $C_\ell(n)$ is true for exactly one object n in this chain. The distance of n from $\mathbf{0}$ in the chain represents the value of the counter c_ℓ. Furthermore, M is in a halting configuration if Halt is true and J_i is the currently executed instruction if the corresponding fluent J_i holds true. The initial theory is given by

$$\begin{aligned} \mathcal{D}_0 := \{ & \forall x.\big(x = \mathbf{0} \equiv C_0(x)\big), \forall x.\big(x = \mathbf{0} \equiv C_1(x)\big), \neg \mathit{Halt}, J_0, \neg J_1, \ldots, \neg J_m, \\ & \forall x.\exists^{=1} y.\mathit{Adj}(x, y), \forall x.\big(x \neq \mathbf{0} \supset \exists^{=1} y.\mathit{Adj}(y, x)\big), \forall x.\neg \mathit{Adj}(x, \mathbf{0})\}. \end{aligned}$$

We use 0-ary actions $\mathit{Inc}_0, \mathit{Inc}_1, \mathit{Dec}_0, \mathit{Dec}_1$, $\mathit{Jump}_0, \ldots \mathit{Jump}_m$ and Stop. For each fluent there is an SSA in $\mathcal{D}_{\text{post}}$. The effect conditions for the fluent $C_\ell(x)$ are given as follows:

$$\begin{aligned} \gamma^{+}_{C_\ell} &:= a = \mathit{Inc}_\ell \wedge \exists y.\big(C_\ell(y) \wedge \mathit{Adj}(y, x)\big) \vee a = \mathit{Dec}_\ell \wedge \exists y.\big(C_\ell(y) \wedge \mathit{Adj}(x, y)\big) \\ \gamma^{-}_{C_\ell} &:= a = \mathit{Inc}_\ell \wedge C_\ell(x) \vee a = \mathit{Dec}_\ell \wedge C_\ell(x). \end{aligned} \tag{3}$$

And for J_j and Halt the positive and negative effect conditions are defined by

$$\begin{gathered} \gamma^{+}_{J_j} := a = \mathit{Jump}_j \text{ and } \gamma^{-}_{J_j} := \bigvee_{j' \neq j} a = \mathit{Jump}_{j'} \\ \gamma^{+}_{\mathit{Halt}} := a = \mathit{Stop} \text{ and } \gamma^{-}_{\mathit{Halt}} := a = a \wedge \mathbf{0} \neq \mathbf{0}. \end{gathered} \tag{4}$$

For each instruction J_j of M we define a program expression δ_j as follows. If $\mathsf{J}_j = \mathsf{Inc}(\ell, i)$, then

$$\delta_j := Inc_\ell; Jump_i.$$

If $\mathsf{J}_j = \mathsf{Dec}(\ell, i, j)$, then

$$\delta_j := \big(C_\ell(\mathbf{0})?; Jump_i\big) \mid \big(\neg C_\ell(\mathbf{0})?; Dec_\ell; Jump_j\big).$$

And if $\mathsf{J}_j = \mathsf{Halt}$, then $\delta_j = Stop$. Now we can assemble the program expression for M.

$$\delta_{\mathsf{M}} := \big(J_0?; \delta_0 \mid \cdots \mid J_m?; \delta_m\big)^*.$$

It is straightforward to show that the temporal state formula $\mathsf{EF}Halt$ is valid in

$$\mathcal{G}_{\mathsf{M}} = (\mathcal{D}_{\mathsf{M}} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}, \delta_{\mathsf{M}})$$

iff M halts. □

3.2 Decidable Verification with Acyclic Action Theories

To achieve decidability of the verification problem we restrict the syntax of the SSAs in the action theory.

Fluent dependencies and acyclic basic action theories

To analyze the source of undecidability we investigate the dependencies between the different fluents occurring in the action theory.

Definition 13. Let $\mathcal{D}$ be a C^2-BAT. The *fluent dependency graph* for $\mathcal{D}$, denoted by $G_{\mathcal{D}}$, consists of a set of nodes, one for each fluent in $\mathcal{D}$. There is a directed edge (F, F') from fluent F to fluent F' iff there exists a disjunct $\exists \vec{y}.\big(a = A(\vec{v}) \wedge \phi \wedge \phi'\big)$ in γ_F^+ or γ_F^- such that F' occurs in the effect descriptor ϕ. We call $\mathcal{D}$ *acyclic* iff $G_{\mathcal{D}}$ is acyclic. The *fluent depth of an acyclic action theory* $\mathcal{D}$, denoted by $\mathsf{fd}(\mathcal{D})$, is given by the length of the longest path in $G_{\mathcal{D}}$. For a fluent F in an acylic BAT $\mathcal{D}$ the *fluent depth of* F *w.r.t.* $\mathcal{D}$, denoted by $\mathsf{fd}_{\mathcal{D}}(F)$, is given by the length of the longest path in $G_{\mathcal{D}}$ starting in F. ▲

Example 14. First, consider the BAT in the undecidability proof. Obviously, the dependency graph is cyclic as there are edges (C_0, C_0) and (C_1, C_1).

On the other hand, the BAT from Example 6 has an acyclic dependency graph (with fluent depth 2) as shown in Figure 4. Fluents *Ovl*, *Server* and *VM* were omitted as they are not incident to any edges. *Ovl* for instance only occurs in the context conditions of γ^+_{Avail}, γ^+_{Hosts} and γ^-_{Hosts}, and *Hosts* in the context condition of γ^+_{Ovl}. For the dependency graph however, only effect descriptors are relevant. For instance, there is an edge from *Avail* to *Runs* because *Runs* occurs in the effect descriptor in conjunction with the migration action in γ^+_{Avail}, i.e. the migration of a virtual machine may affects the availability of all processes running on this machine. In an analogous way *Avail* and *Hosts*, *Proc* are related due to the effect descriptor of the repair action in γ^+_{Avail}. The other edges can be explained similarly. ▲

Note that if actions have only *local-effects* [VLL08], then $\mathcal{D}$ is acyclic. In case of local-effect actions the effect descriptors do not contain any fluents. Consequently, the corresponding BAT has fluent depth 0. Another well-known special case are context-free actions [LR97] where the positive and negative effect conditions are restricted to contain only *rigid predicate symbols*. Clearly, BATs restricted in this way have at most fluent depth 1. The so called *solitary stratified theories* considered in [McI00] are based on a similar acyclicity condition, but without distinguishing between effect descriptors and context conditions. The action theory in our example is therefore not a solitary stratified theory.

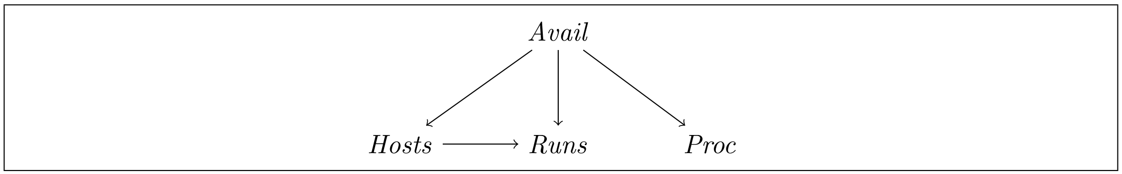

Figure 4: Example fluent dependencies

Compact representation of effects using regression

We restrict our attention to programs $\mathcal{G} = (\mathcal{D}, \delta)$ with an acyclic C^2-BAT $\mathcal{D}$. The finite set of ground actions (including ϵ and $\mathfrak{f}$) is denoted by $\mathcal{A}$ and the finite set of fluents in $\mathcal{G} = (\mathcal{D}, \delta)$ by $\mathcal{F}$ in the following. We construct finite propositional abstractions of the transition systems T_δ^w with $w \models \mathcal{D}$. The essential part for this abstraction is a compact representation of the effects generated by executing a *sequence* of ground actions in a given world satisfying the BAT.

In case of local-effect actions as considered in [ZC14] the idea is as follows. The execution of a ground action $A(\vec{c})$ with only local-effects changes only the truth values of primitive formulas of the form $F(\vec{n})$ where the objects $\vec{n}$ are arguments of the action, i.e. they are contained in $\vec{c}$. Therefore, to capture the effects of any action sequence admitted by a program over local-effect actions it is sufficient to consider all finitely many finite sets of fluent literals built from the fluents and objects mentioned in the program.

Since a ground action defined in an acyclic BAT may change the truth values of possibly infinitely many primitive formulas, a direct adaption of the approach for local-effect actions is not possible.

First, we consider the *ground instantiations of the SSAs* where the universally quantified action variable a is substituted with actions from $\mathcal{A}$.

Let $F(\vec{x}) \in \mathcal{F}$ and $t \in \mathcal{A}$. The ground instantiation of the SSA of F with t is of the form

$$\Box[t]F(\vec{x}) \equiv \big(\gamma_F^+\big)_t^a \vee F(\vec{x}) \wedge \neg\big(\gamma_F^-\big)_t^a.$$

Lemma 15. *The instantiated positive and negative effect conditions* $\big(\gamma_F^+\big)_t^a$ *and* $\big(\gamma_F^-\big)_t^a$ *are, respectively equivalent to a disjunction of the form*

$$\phi_1^{\mathsf{eff}} \wedge \phi_1^{\mathsf{con}} \vee \dots \vee \phi_n^{\mathsf{eff}} \wedge \phi_n^{\mathsf{con}} \tag{5}$$

for some $n \geq 0$, *where the formulas* $\phi_1^{\mathsf{eff}}, \dots, \phi_n^{\mathsf{eff}}$, *are* C^2*-fluent formulas with* $\vec{x}$ *as free variables and no other free variables and the formulas* $\phi_1^{\mathsf{con}}, \dots, \phi^{\mathsf{con}}$ *are* C^2*-fluent sentences.*

Proof. According to Definition 5 of C^2-BATs the effect conditions are disjunctions of formulas of the form

$$\exists\vec{y}.\big(a = A(\vec{v}) \wedge \phi \wedge \phi'\big). \tag{6}$$

After replacing a with a ground action term t the disjunct is either equivalent to false or there is a matcher for the variables in $\vec{v}$ and the free variables in ϕ and ϕ' can be replaced by constants such that ϕ^{eff} corresponds to the effect descriptor and ϕ^{con} to the context condition ϕ'. We can proceed with all disjuncts in $\big(\gamma_F^+\big)_t^a$ and $\big(\gamma_F^-\big)_t^a$ in this way and obtain an equivalent formula of the form (5). □

In the remainder of this report we assume that for each $F(\vec{x}) \in \mathcal{F}$ and $t \in \mathcal{A}$ the instantiated positive and negative effect conditions $\big(\gamma_F^+\big)_t^a$ and $\big(\gamma_F^-\big)_t^a$ are, respectively, given as a fixed disjunction of the form (5). We call the formulas $\phi_1^{\mathsf{eff}}, \ldots, \phi_n^{\mathsf{eff}}$ *effect descriptors* and $\phi_1^{\mathsf{con}}, \ldots, \phi_n^{\mathsf{con}}$ *context conditions.*

In the following we often view the instantiated effect condition $\left(\gamma_F^+\right)_t^a$ and $\left(\gamma_F^-\right)_t^a$, respectively, as a set of the form $\{(\phi_1^{\mathsf{eff}}, \phi_1^{\mathsf{con}}), \ldots, (\phi_n^{\mathsf{eff}}, \phi_n^{\mathsf{con}})\}$. We write $(\phi_i^{\mathsf{eff}}, \phi_i^{\mathsf{con}}) \in \left(\gamma_F^+\right)_t^a$ and $(\phi_i^{\mathsf{eff}}, \phi_i^{\mathsf{con}}) \in \left(\gamma_F^-\right)_t^a$ to denote that $\phi_i^{\mathsf{eff}} \wedge \phi_i^{\mathsf{con}}$ is a disjunct in $\left(\gamma_F^+\right)_t^a$ and $\left(\gamma_F^-\right)_t^a$, respectively.

For a given fluent $F \in \mathcal{F}$ and set of ground action $\mathcal{A}$ we define the *sets of relevant effect descriptors* as follows:

$$\begin{aligned} \mathsf{eff}_{\mathcal{A}}^+(F) &:= \{\phi^{\mathsf{eff}} \mid (\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^+\right)_t^a \text{ for some } t \in \mathcal{A}\} \\ \mathsf{eff}_{\mathcal{A}}^-(F) &:= \{\phi^{\mathsf{eff}} \mid (\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^-\right)_t^a \text{ for some } t \in \mathcal{A}\}. \end{aligned} \tag{7}$$

First, we introduce an action-centric representation of effects that captures also a possible unbounded number of affected primitive formulas.

Definition 16. Let $F(\vec{x})$ be a fluent and ϕ a C^2-fluent formula with free variables $\vec{x}$. We call the expression $\langle F^+, \phi\rangle$ a *positive effect on* F, and the expression $\langle F^-, \phi\rangle$ is called a *negative effect on* F. We use the notation $\langle F^\pm, \phi\rangle$ if we do not explicitly distinguish between a positive or negative effect on F.

Let $\mathcal{D}$ be a C^2-BAT, w a world with $w \models \mathcal{D}$, $z \in \mathcal{Z}$ and $t \in \mathcal{A}$. The *effects of executing* t *in* (w, z) are defined as follows.

$$\begin{aligned} \mathcal{E}_{\mathcal{D}}(w, z, t) :=& \{\langle F^+, \phi^{\mathsf{eff}}\rangle \mid \exists (\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^+\right)_t^a \text{ such that } w, z \models \phi^{\mathsf{con}}\} \cup \\ & \{\langle F^-, \phi^{\mathsf{eff}}\rangle \mid \exists (\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^-\right)_t^a \text{ such that } w, z \models \phi^{\mathsf{con}}\}. \end{aligned}$$

▲

Intuitively, if $\langle F^+, \phi\rangle \in \mathcal{E}_{\mathcal{D}}(w, z, t)$ and $w, z \models \phi_{\vec{c}}^{\vec{x}}$ holds *before* executing t in w, z, then $F(\vec{c})$ will be true *after* the execution. Likewise, if $\langle F^-, \phi\rangle \in \mathcal{E}_{\mathcal{D}}(w, z, t)$ and $w, z \models \phi_{\vec{c}}^{\vec{x}}$ holds *before* executing t in w, z, then $F(\vec{c})$ will be false *after* the execution.

To accumulate the effects of consecutively executed actions we define a regression operator applied to a C^2-fluent formula given a set of effects. From now on we assume that only the object variable symbols x and y are used in C^2-fluent formulas. For a given fluent formula ϕ, the formula $\widehat{\phi}$ is obtained from ϕ by replacing each occurrence (bound and free) of x in ϕ by y and each occurrence of y by x.

Definition 17. Let E be a set of effects and φ a C^2-fluent formula. The *regression of* φ *through* E, denoted by $\mathcal{R}[\mathsf{E}, \varphi]$, is a C^2-fluent formula defined by induction on the structure of φ as given in Figure 5. If a set of effects E contains no effect on F, then $\mathcal{R}[\mathsf{E}, F(\vec{v})] = F(\vec{v})$. And if E is the empty set, then we have $\mathcal{R}[\mathsf{E}, \phi] = \phi$ for any C^2-fluent formula ϕ. Note that as usual we assume that the empty disjunction is *false* and the empty conjunction is *true*. ▲

We show a standard property of the one-step regression operator.

Lemma 18. *Let* $\mathcal{D}$ *be a* C^2*-BAT and* $w \in \mathcal{W}$ *such that* $w \models \mathcal{D}$, $z \in \mathcal{Z}$, $t \in \mathcal{A}$ *and* ψ *a* C^2*-fluent sentence. It holds that*

$$w, z \cdot t \models \psi \textit{ iff } w, z \models \mathcal{R}[\mathsf{E}, \psi] \textit{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).$$

Proof. The proof is done by structural induction on fluent formulas using the definition of the regression operator and the definition of action effects. Let $w \in \mathcal{W}$ with $w \models \mathcal{D}$, $z \in \mathcal{Z}$, $t \in \mathcal{A}$,

$$
\mathcal{R}[\mathsf{E}, F(v)] := \begin{cases} F(v) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}} \neg \varphi^{x}_{v} \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}} \varphi^{x}_{v} & v \neq y \\ F(v) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}} \neg \widehat{\varphi} \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}} \widehat{\varphi} & v = y \end{cases}
$$

$$
\mathcal{R}[\mathsf{E}, F(v_1, v_2)] := \begin{cases} F(v_1, v_2) \wedge \bigwedge_{\langle F(x,y)^{-}, \varphi\rangle \in \mathsf{E}} \exists y.\big(x = y \wedge \neg\varphi\big) \vee & v_1 = x, v_2 = x \\ \quad \bigvee_{\langle F(x,y)^{+}, \varphi\rangle \in \mathsf{E}} \exists y.\big(x = y \wedge \varphi\big) & \\ F(v_1, v_2) \wedge \bigwedge_{\langle F(x,y)^{-}, \varphi\rangle \in \mathsf{E}} \exists x.\big(y = x \wedge \neg\varphi\big) \vee & v_1 = y, v_2 = y \\ \quad \bigvee_{\langle F(x,y)^{+}, \varphi\rangle \in \mathsf{E}} \exists x.\big(y = x \wedge \varphi\big) & \\ F(v_1, v_2) \wedge \bigwedge_{\langle F(x,y)^{-}, \varphi\rangle \in \mathsf{E}} \neg\big(\widehat{\varphi}\big)^{x\ y}_{v_2\ v_1} \vee \bigvee_{\langle F(x,y)^{+}, \varphi\rangle \in \mathsf{E}} \big(\widehat{\varphi}\big)^{x\ y}_{v_2\ v_1} & \begin{array}{l} v_1 = y, v_2 = x \text{ or} \\ v_1 = y, v_2 = c \text{ or} \\ v_1 = c, v_2 = x \end{array} \\ F(v_1, v_2) \wedge \bigwedge_{\langle F(x,y)^{-}, \varphi\rangle \in \mathsf{E}} \neg\big(\varphi\big)^{x\ y}_{v_1\ v_2} \vee \bigvee_{\langle F(x,y)^{+}, \varphi\rangle \in \mathsf{E}} \big(\varphi\big)^{x\ y}_{v_1\ v_2} & \text{otherwise} \end{cases}
$$

$$
\begin{aligned}
\mathcal{R}[\mathsf{E}, t_1 = t_2] &:= t_1 = t_2 \\
\mathcal{R}[\mathsf{E}, \phi_1 \wedge \phi_2] &:= \mathcal{R}[\mathsf{E}, \phi_1] \wedge \mathcal{R}[\mathsf{E}, \phi_2] \\
\mathcal{R}[\mathsf{E}, \forall x.\phi] &:= \forall x.\mathcal{R}[\mathsf{E}, \phi] \\
\mathcal{R}[\mathsf{E}, \exists^{\leq m} x.\phi] &:= \exists^{\leq m} x.\mathcal{R}[\mathsf{E}, \phi].
\end{aligned}
$$

Figure 5: Regression operator adapted from [GS07]

ϕ a C^2-fluent formula where $\vec{v}$ are the free variables in ϕ and let $\vec{c}$ be a sequence of object standard names of the same length as $\vec{v}$.

$$w, z \cdot t \models \phi^{\vec{v}}_{\vec{c}} \text{ iff } w, z \models \big(\mathcal{R}[\mathsf{E}, \phi]\big)^{\vec{v}}_{\vec{c}} \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).$$

First assume $\phi = F(x)$. The ground instantiated SSA for F with t in $\mathcal{D}$ is of the form

$$\Box[t]F(x) \equiv \big(\gamma_F^{+}\big)^a_t \vee F(x) \wedge \neg\big(\gamma_F^{-}\big)^a_t \tag{8}$$

where the effect conditions $\big(\gamma_F^{+}\big)^a_t$ and $\big(\gamma_F^{-}\big)^a_t$ are given by

$$\begin{aligned} \big(\gamma_F^{+}\big)^a_t &= \phi_1^{\mathsf{eff}} \wedge \phi_1^{\mathsf{con}} \vee \cdots \vee \phi_n^{\mathsf{eff}} \wedge \phi_n^{\mathsf{con}} \text{ and} \\ \big(\gamma_F^{-}\big)^a_t &= \varphi_1^{\mathsf{eff}} \wedge \varphi_1^{\mathsf{con}} \vee \cdots \vee \varphi_m^{\mathsf{eff}} \wedge \varphi_m^{\mathsf{con}}. \end{aligned} \tag{9}$$

It holds that $w, z \cdot t \models \big(F(x)\big)^x_c$ iff $w, z \cdot t \models F(c)$

$$\begin{aligned}
&\text{iff } w, z \models [t]F(c) \\
&\text{iff } w, z \models \Big(\big(\gamma_F^{+}\big)^a_t \vee F(x) \wedge \neg\big(\gamma_F^{-}\big)^a_t\Big)^x_c \text{ (since } w \models \mathcal{D}) \\
&\text{iff } w, z \models \big(\phi_1^{\mathsf{eff}} \wedge \phi_1^{\mathsf{con}} \vee \cdots \vee \phi_n^{\mathsf{eff}} \wedge \phi_n^{\mathsf{con}}\big)^x_c \vee \\
&\qquad\qquad F(c) \wedge \neg\big(\varphi_1^{\mathsf{eff}} \wedge \varphi_1^{\mathsf{con}} \vee \cdots \vee \varphi_m^{\mathsf{eff}} \wedge \varphi_m^{\mathsf{con}}\big)^x_c \text{ (with (9))} \\
&\text{iff } w, z \models \big(\phi_1^{\mathsf{eff}}\big)^x_c \wedge \phi_1^{\mathsf{con}} \vee \cdots \vee \big(\phi_n^{\mathsf{eff}}\big)^x_c \wedge \phi_n^{\mathsf{con}} \vee \\
&\qquad\qquad F(c) \wedge \Big(\neg\big(\varphi_1^{\mathsf{eff}}\big)^x_c \vee \neg\varphi_1^{\mathsf{con}}\Big) \wedge \cdots \wedge \Big(\neg\big(\varphi_m^{\mathsf{eff}}\big)^x_c \vee \neg\varphi_m^{\mathsf{con}}\Big) \\
&\text{iff } w, z \models \bigvee_{\substack{\phi_i^{\mathsf{con}} \in \{\phi_1^{\mathsf{con}}, \ldots, \phi_n^{\mathsf{con}}\}, \\ w, z \models \phi_i^{\mathsf{con}}}} \big(\phi_i^{\mathsf{eff}}\big)^x_c \vee F(c) \wedge \bigwedge_{\substack{\varphi_i^{\mathsf{con}} \in \{\varphi_1^{\mathsf{con}}, \ldots, \varphi_m^{\mathsf{con}}\}, \\ w, z \models \varphi_i^{\mathsf{con}}}} \neg\big(\varphi_i^{\mathsf{eff}}\big)^x_c \\
&\text{iff } w, z \models \bigvee_{\substack{(\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^{+}\right)^a_t, \\ w, z \models \phi^{\mathsf{con}}}} \big(\phi^{\mathsf{eff}}\big)^x_c \vee F(c) \wedge \bigwedge_{\substack{(\varphi^{\mathsf{eff}}, \varphi^{\mathsf{con}}) \in \left(\gamma_F^{-}\right)^a_t, \\ w, z \models \varphi^{\mathsf{con}}}} \neg\big(\varphi^{\mathsf{eff}}\big)^x_c \\
&\text{iff } w, z \models \bigvee_{\langle F(x)^{+}, \phi\rangle \in \mathsf{E}} \phi^x_c \vee F(c) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}} \neg\varphi^x_c \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t) \\
&\text{iff } w, z \models \big(\mathcal{R}[\mathsf{E}, F(x)]\big)^x_c \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).
\end{aligned}$$

Next consider the case $\phi = F(y)$. It holds that $w, z \cdot t \models \big(F(y)\big)^y_c$ iff $w, z \cdot t \models F(c)$. As shown above it holds that

$$w, z \cdot t \models F(c) \text{ iff } w, z \models \bigvee_{\langle F(x)^{+}, \phi\rangle \in \mathsf{E}} \phi^x_c \vee F(c) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}} \neg\varphi^x_c \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).$$

Obviously, for an effect formula ϕ with free variable x it holds that $w, z \models \phi^x_c$ iff $w, z \models \widehat{\phi}^y_c$, where $\widehat{\phi}$ is obtained from ϕ by swapping the variable symbols x and y. By definition of the regression operator we therefore obtain

$$w, z \cdot t \models \big(F(y)\big)^y_c \text{ iff } w, z \models \big(\mathcal{R}[\mathsf{E}, F(y)]\big)^y_c \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).$$

Next consider the case $\phi = F(x, x)$. The ground instantiated SSA for F is of the form

$$\Box[t]F(x, y) \equiv \big(\gamma_F^{+}\big)^a_t \vee F(x, y) \wedge \neg\big(\gamma_F^{-}\big)^a_t. \tag{10}$$

There are effect formulas $\phi_1^{\mathsf{eff}}, \ldots, \phi_n^{\mathsf{eff}}$ and $\varphi_1^{\mathsf{eff}}, \ldots, \varphi_m^{\mathsf{eff}}$ with x and y as the free variables such that

$$\begin{aligned}\big(\gamma_F^+\big)^a_t &= \phi_1^{\mathsf{eff}} \land \phi_1^{\mathsf{con}} \lor \cdots \lor \phi_n^{\mathsf{eff}} \land \phi_n^{\mathsf{con}} \text{ and}\\ \big(\gamma_F^-\big)^a_t &= \varphi_1^{\mathsf{eff}} \land \varphi_1^{\mathsf{con}} \lor \cdots \lor \varphi_m^{\mathsf{eff}} \land \varphi_m^{\mathsf{con}}.\end{aligned} \tag{11}$$

It holds that $w, z \cdot t \models \big(F(x,x)\big)^x_c$ iff $w, z \cdot t \models F(c,c)$

iff $w, z \models [t]F(c,c)$

iff $w, z \models \Big(\big(\gamma_F^+\big)^a_t \lor F(x,y) \land \neg\big(\gamma_F^-\big)^a_t\Big)^{x\,y}_{c\,c}$ (since $w \models \mathcal{D}$)

iff $w, z \models \big(\phi_1^{\mathsf{eff}} \land \phi_1^{\mathsf{con}} \lor \cdots \lor \phi_n^{\mathsf{eff}} \land \phi_n^{\mathsf{con}}\big)^{x\,y}_{c\,c} \lor$
$F(c,c) \land \neg\big(\varphi_1^{\mathsf{eff}} \land \varphi_1^{\mathsf{con}} \lor \cdots \lor \varphi_m^{\mathsf{eff}} \land \varphi_m^{\mathsf{con}}\big)^{x\,y}_{c\,c}$ (with (9))

iff $w, z \models \big(\phi_1^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \land \phi_1^{\mathsf{con}} \lor \cdots \lor \big(\phi_n^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \land \phi_n^{\mathsf{con}} \lor$
$F(c,c) \land \Big(\neg\big(\varphi_1^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \lor \neg\varphi_1^{\mathsf{con}}\Big) \land \cdots \land \Big(\neg\big(\varphi_m^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \lor \neg\varphi_m^{\mathsf{con}}\Big)$

iff $w, z \models \bigvee_{\substack{\phi_i^{\mathsf{con}} \in \{\phi_1^{\mathsf{con}}, \ldots, \phi_n^{\mathsf{con}}\},\\ w,z \models \phi_i^{\mathsf{con}}}} \big(\phi_i^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \lor F(c,c) \land \bigwedge_{\substack{\varphi_i^{\mathsf{con}} \in \{\varphi_1^{\mathsf{con}}, \ldots, \varphi_m^{\mathsf{con}}\},\\ w,z \models \varphi_i^{\mathsf{con}}}} \neg\big(\varphi_i^{\mathsf{eff}}\big)^{x\,y}_{c\,c}$

iff $w, z \models \bigvee_{\substack{(\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in (\gamma_F^+)^a_t,\\ w,z \models \phi^{\mathsf{con}}}} \big(\phi^{\mathsf{eff}}\big)^{x\,y}_{c\,c} \lor F(c,c) \land \bigwedge_{\substack{(\varphi^{\mathsf{eff}}, \varphi^{\mathsf{con}}) \in (\gamma_F^-)^a_t,\\ w,z \models \varphi^{\mathsf{con}}}} \neg\big(\varphi^{\mathsf{eff}}\big)^{x\,y}_{c\,c}$

iff $w, z \models \bigvee_{\langle F(x,y)^+, \phi\rangle \in \mathsf{E}} \phi^{x\,y}_{c\,c} \lor F(c,c) \land \bigwedge_{\langle F(x,y)^-, \varphi\rangle \in \mathsf{E}} \neg\varphi^{x\,y}_{c\,c}$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$

iff $w, z \models \bigvee_{\langle F(x,y)^+, \phi\rangle \in \mathsf{E}} \big(\exists y. x = y \land \phi\big)^x_c \lor F(c,c) \land \bigwedge_{\langle F(x,y)^-, \varphi\rangle \in \mathsf{E}} \big(\exists y. x = y \land \neg\varphi\big)^x_c$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$ since for a fluent formula α with free variables x and y it obviously holds that

$$w, z \models \alpha^{x\,y}_{c\,c} \text{ iff } \big(\exists y. x = y \land \alpha\big)^x_c.$$

iff $w, z \models \big(\mathcal{R}[\mathsf{E}, F(x,x)]\big)^x_c$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$.

We omit the cases $\phi = F(y,y)$, $F(y,x)$, $F(x,y)$. They follow with analogous arguments.

Now let $\phi = (t_1 = t_2)$ with free variables $\vec{v}$. We have $(t_1 = t_2)^{\vec{v}}_{\vec{c}} = (n_1 = n_2)$ where n_1 and n_2 are standard names. The claim follows immediately.

Let $\phi = \varphi_1 \land \varphi_2$. It holds that $w, z \cdot t \models \big(\varphi_1 \land \varphi_2\big)^{\vec{v}}_{\vec{c}}$

iff $w, z \cdot t \models \big(\varphi_1\big)^{\vec{v}}_{\vec{c}}$ and $w, z \cdot t \models \big(\varphi_2\big)^{\vec{v}}_{\vec{c}}$

iff $w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi_1]\big)^{\vec{v}}_{\vec{c}}$ and $w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi_2]\big)^{\vec{v}}_{\vec{c}}$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$ (by induction)

iff $w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi_1]\big)^{\vec{v}}_{\vec{c}} \land \big(\mathcal{R}[\mathsf{E}, \varphi_2]\big)^{\vec{v}}_{\vec{c}}$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$

iff $w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi_1] \land \mathcal{R}[\mathsf{E}, \varphi_2]\big)^{\vec{v}}_{\vec{c}}$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$

iff $w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi_1 \land \varphi_2]\big)^{\vec{v}}_{\vec{c}}$ with $\mathsf{E} = \mathcal{E}_{\mathcal{D}}(w,z,t)$.

Let $\phi = \forall x.\varphi$. Obviously, y is the only free variable in $\forall x.\varphi$ or there is no free variable. It holds that $w, z \cdot t \models \big(\forall x.\varphi\big)^{y}_{c}$

$$\begin{aligned}
&\text{iff } w, z \cdot t \models \forall x.\big(\varphi\big)^{y}_{c}\\
&\text{iff } w, z \cdot t \models \big(\varphi\big)^{y\,x}_{c\,n} \text{ for all } n \in \mathcal{N}_O\\
&\text{iff } w, z \models \big(\mathcal{R}[\mathsf{E}, \varphi]\big)^{y\,x}_{c\,n} \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t) \text{ for all } n \in \mathcal{N}_O \text{ (by induction)}\\
&\text{iff } w, z \models \forall x.\big(\mathcal{R}[\mathsf{E}, \varphi]\big)^{y}_{c} \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t)\\
&\text{iff } w, z \models \big(\forall x.\mathcal{R}[\mathsf{E}, \varphi]\big)^{y}_{c} \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t)\\
&\text{iff } w, z \models \big(\mathcal{R}[\mathsf{E}, \forall x.\varphi]\big)^{y}_{c} \text{ with } \mathsf{E} = \mathcal{E}_{\mathcal{D}}(w, z, t).
\end{aligned}$$

We omit the cases $\phi = \exists^{\leq m} x.\varphi$ and $\phi = \exists^{\geq m} x.\varphi$. They can be shown with analogous arguments. □

Using the regression operator we can accumulate several sets of effects. Let E_0 and E_1 be two sets of effects. First executing E_0 and then afterwards E_1 yields a set of effects, denoted by $\mathsf{E}_0 \rhd \mathsf{E}_1$, that is defined as follows:

$$\begin{aligned}
\mathsf{E}_0 \rhd \mathsf{E}_1 :=& \{\langle F^{\pm}, \mathcal{R}[\mathsf{E}_0, \varphi]\rangle \mid \langle F^{\pm}, \varphi\rangle \in \mathsf{E}_1\} \cup\\
& \{\langle F^{+}, (\varphi \wedge \bigwedge_{\langle F^{-}, \varphi'\rangle \in \mathsf{E}_1} \neg\mathcal{R}[\mathsf{E}_0, \varphi'])\rangle \mid \langle F^{+}, \varphi\rangle \in \mathsf{E}_0\} \cup\\
& \{\langle F^{-}, \varphi\rangle \in \mathsf{E}_0\}.
\end{aligned} \tag{12}$$

Lemma 19. *Let ϕ be a C^2-fluent formula and E_0 and E_1 two sets of effects. It holds that $\mathcal{R}[\mathsf{E}_0, \mathcal{R}[\mathsf{E}_1, \phi]] \equiv \mathcal{R}[\mathsf{E}_0 \rhd \mathsf{E}_1, \phi]$.*

Proof. Let E_0, E_1 be sets of effects and ϕ a C^2-fluent formula. We show by induction on the structure of ϕ that $\mathcal{R}[\mathsf{E}_0, \mathcal{R}[\mathsf{E}_1, \phi]] \equiv \mathcal{R}[\mathsf{E}_0 \rhd \mathsf{E}_1, \phi]$. We consider the case $\phi = F(v)$ with $v \neq y$.

It holds that $\mathcal{R}[\mathsf{E}_0, \mathcal{R}[\mathsf{E}_1, F(v)]]$

$$\begin{aligned}
=\;& \mathcal{R}[\mathsf{E}_0, \Bigg(F(v) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}_1} \neg\varphi^{x}_{v}\Bigg) \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}_1} \varphi^{x}_{v}]\\
=\;& \Bigg(\mathcal{R}[\mathsf{E}_0, F(v)] \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}_1} \neg\mathcal{R}[\mathsf{E}_0, \varphi^{x}_{v}]\Bigg) \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}_1} \mathcal{R}[\mathsf{E}_0, \varphi^{x}_{v}]\\
\equiv\;& \Bigg(\mathcal{R}[\mathsf{E}_0, F(v)] \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}_1} \neg\big(\mathcal{R}[\mathsf{E}_0, \varphi]\big)^{x}_{v}\Bigg) \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}_1} \big(\mathcal{R}[\mathsf{E}_0, \varphi]\big)^{x}_{v}\\
=\;& \Bigg(\Bigg(F(v) \wedge \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}_0} \neg\varphi^{x}_{v} \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}_0} \varphi^{x}_{v}\Bigg) \wedge\\
& \bigwedge_{\langle F(x)^{-}, \varphi\rangle \in \mathsf{E}_1} \neg\big(\mathcal{R}[\mathsf{E}_0, \varphi]\big)^{x}_{v}\Bigg) \vee \bigvee_{\langle F(x)^{+}, \varphi\rangle \in \mathsf{E}_1} \big(\mathcal{R}[\mathsf{E}_0, \varphi]\big)^{x}_{v}
\end{aligned}$$

$$\equiv \Bigg(\Bigg(F(v) \wedge \bigwedge_{\langle F(x)^-,\varphi\rangle\in\mathsf{E}_0} \Big(\neg\varphi^x_v \wedge \bigwedge_{\langle F(x)^-,\varphi\rangle\in\mathsf{E}_1} \neg\big(\mathcal{R}[\mathsf{E}_0,\varphi]\big)^x_v\Big)\Bigg) \vee$$

$$\bigvee_{\langle F(x)^+,\varphi\rangle\in\mathsf{E}_0} \Big(\varphi^x_v \wedge \bigwedge_{\langle F(x)^-,\varphi\rangle\in\mathsf{E}_1} \neg\big(\mathcal{R}[\mathsf{E}_0,\varphi]\big)^x_v\Big)\Bigg) \vee \bigvee_{\langle F(x)^+,\varphi\rangle\in\mathsf{E}_1} \big(\mathcal{R}[\mathsf{E}_0,\varphi]\big)^x_v$$

$$\equiv \mathcal{R}[\mathsf{E}_0 \rhd \mathsf{E}_1, F(v)].$$

We omit the remaining cases here. They can be shown following the same lines. □

We can now accumulate the effects of a sequence of actions into a single set of effects.

Definition 20. Let $\mathcal{D}$ be a C^2-BAT, $\mathcal{A}$ a finite set of ground actions, w a world with $w \models \mathcal{D}$, and $z = t_1t_2\cdots t_n \in \mathcal{A}^*$ a sequence of ground actions of length $n \in \mathbb{N}$. For a given $i \le n$, $z[i]$ denotes the subsequence of z that consists of the first i elements of z. We define the following sequence of sets of effects:

$$\begin{aligned}\mathsf{E}_1 &:= \mathcal{E}_{\mathcal{D}}(w, \langle\rangle, t_1)\\ \mathsf{E}_i &:= \mathsf{E}_{i-1} \rhd \mathcal{E}_{\mathcal{D}}(w, z[i-1], t_i) \text{ for } i = 2,\ldots,n.\end{aligned}$$

We say that E_n *is generated by executing* $t_1t_2\cdots t_n$ *in* w. ▲

We can now generalize Lemma 18 to the case with a sequence of ground actions.

Lemma 21. *Let $\mathcal{D}$, $w \models \mathcal{D}$, $z \in \mathcal{A}^*$ be as above. For the effects E_z generated by executing z in w and a C^2-fluent sentence ψ it holds that $w, z \models \psi$ iff $w, \langle\rangle \models \mathcal{R}[\mathsf{E}_z, \psi]$.*

Proof. This lemma is a direct consequence of Lemma 18 and Lemma 19. □

We can now define a finite representation of all effects that can be generated with actions from $\mathcal{A}$ defined in an acyclic BAT.

Definition 22 (relevant effects)**.** Let $\mathcal{D}$ be an acyclic BAT w.r.t. $\mathcal{A}$ with $\mathsf{fd}(\mathcal{D}) = n$. We define a sequence of sets of effects, denoted by $\mathfrak{E}^{\mathcal{D},\mathcal{A}}_0, \mathfrak{E}^{\mathcal{D},\mathcal{A}}_1, \ldots, \mathfrak{E}^{\mathcal{D},\mathcal{A}}_n$, as follows:

$$\begin{aligned}\mathfrak{E}^{\mathcal{D},\mathcal{A}}_0 &:= \{\langle F^{\pm}, \varphi\rangle \mid \mathsf{fd}_{\mathcal{D}}(F) = 0, \varphi \in \mathsf{eff}^+_{\mathcal{A}}(F) \cup \mathsf{eff}^-_{\mathcal{A}}(F)\};\\ \mathfrak{E}^{\mathcal{D},\mathcal{A}}_i &:= \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1} \cup \{\langle F^-, \mathcal{R}[\mathsf{E},\varphi]\rangle \mid \mathsf{fd}_{\mathcal{D}}(F) = i, \varphi \in \mathsf{eff}^-_{\mathcal{A}}(F), \mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}\} \cup\\ &\quad\ \{\langle F^+, \Big(\mathcal{R}[\mathsf{E},\phi] \wedge \bigwedge_{(\varphi,\mathsf{E}')\in X} \neg\mathcal{R}[\mathsf{E}',\varphi]\Big)\rangle \mid \begin{aligned}[t]&\mathsf{fd}_{\mathcal{D}}(F) = i,\\ &\phi \in \mathsf{eff}^+_{\mathcal{A}}(F), \mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}\\ &X \subseteq \big(\mathsf{eff}^-_{\mathcal{A}}(F) \times 2^{\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}}\big)\}.\end{aligned}\\ &\quad\ \text{for } i = 1,\ldots,n.\end{aligned}$$

The *set of all relevant effects w.r.t. $\mathcal{D}$ and $\mathcal{A}$* is given by $\mathfrak{E}^{\mathcal{D},\mathcal{A}} := \mathfrak{E}^{\mathcal{D},\mathcal{A}}_n$. ▲

Obviously, $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$ is a finite set of effects. We show that any set of effects generated by executing an action sequence with actions from $\mathcal{A}$ in a world that satisfies $\mathcal{D}$ is a subset of $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$. For the proof we first need some auxiliary notions and properties.

Lemma 23. *Let $\mathcal{D}$ be an acyclic C^2-BAT, $\mathcal{F}$ the set of fluents occurring in $\mathcal{D}$, $w \in \mathcal{W}$ with $w \models \mathcal{D}$, $z \in \mathcal{A}^*$ and $t \in \mathcal{A}$. It holds that $\mathcal{E}_{\mathcal{D}}(w, z, t) \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$.*

Proof. Let $\mathsf{fd}(\mathcal{D}) = f$. Let $\langle F^{\pm}, \phi\rangle \in \mathcal{E}_{\mathcal{D}}(w, z, t)$ with $\mathsf{fd}_{\mathcal{D}}(F) = i$ for some $i \in \{0, \dots, f\}$. By definition of $\mathcal{E}_{\mathcal{D}}(w, z, t)$ it holds that $\phi \in \mathsf{eff}^{+}_{\mathcal{A}}(F) \cup \mathsf{eff}^{-}_{\mathcal{A}}(F)$ and by definition of $\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i}$ it holds that $\langle F^{\pm}, \phi\rangle \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$. □

Let E be a set of effects on fluents from $\mathcal{F}$ defined in an acyclic BAT $\mathcal{D}$ and $f \in \mathbb{N}$. The restriction of E to fluents of depth $\leq f$ is given by

$$\mathsf{E}_{\leq f} := \{\langle F^{\pm}, \phi\rangle \in \mathsf{E} \mid \mathsf{fd}_{\mathcal{D}}(F) \leq f\}. \tag{13}$$

Let ϕ be a fluent formula built from fluents in $\mathcal{F}$. The *fluent depth of* ϕ, denoted by $\mathsf{fd}_{\mathcal{D}}(\phi)$, is the maximal depth among the depths of all the fluents occurring in ϕ.

Lemma 24. *Let $\mathcal{D}$, $\mathcal{A}$ and $\mathcal{F}$ be as above. Let E be a set of effects on $\mathcal{F}$, ϕ a fluent formula over $\mathcal{F}$ with $\mathsf{fd}_{\mathcal{D}}(\phi) \leq f$. It holds that $\mathcal{R}[\mathsf{E}, \phi] = \mathcal{R}[\mathsf{E}_{\leq f}, \phi]$.*

Proof. It follows from the definition of the regression operator that for the regression result $\mathcal{R}[\mathsf{E}, \phi]$ effects in E on fluents that do not occur in ϕ are irrelevant and can be omitted. Therefore the claim follows immediately. □

We say that two effects $\langle F(\vec{x})^{\pm}, \varphi\rangle$ and $\langle F(\vec{x})^{\pm}, \varphi'\rangle$ are equivalent iff $\varphi \equiv \varphi'$, i.e. $\forall\vec{x}.(\varphi \equiv \varphi')$ is a tautology, and the effects are both positive or both negative. Furthermore, let E and E' be two finite sets of effects. We write $\mathsf{E} \equiv \mathsf{E}'$ iff for each effect in E there is an equivalent effect in E' and vice versa. For equivalent effect sets we need the following lemma.

Lemma 25. *Let E and E' be sets of effects with $\mathsf{E} \equiv \mathsf{E}'$.*

1. *Let φ be a C^2-fluent formula with free variables $\vec{x}$. It holds that $\forall\vec{x}.\big(\mathcal{R}[\mathsf{E}, \varphi] \equiv \mathcal{R}[\mathsf{E}', \varphi]\big)$ is a tautology.*
2. *Let E'' be a set of effects. It holds that $(\mathsf{E} \rhd \mathsf{E}'') \equiv (\mathsf{E}' \rhd \mathsf{E}'')$.*

Finally, we show that $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$ indeed captures all the effects that can be generated with actions from $\mathcal{A}$ in worlds satisfying $\mathcal{D}$. Intuitively, for a given fluent F with $\mathsf{fd}_{\mathcal{D}}(F) = 0$ it holds that either F is rigid, i.e. there are no effects on F, or there are only local effects on F. Consequently, all effects on F generated by a ground action sequence from $\mathcal{A}$ must be contained in $\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{0}$. For fluents F with $\mathsf{fd}_{\mathcal{D}}(F) = i$ and $i > 0$ the fluents in the effect descriptors may also be subject to changes but have a depth strictly smaller than i. To obtain all relevant effects on F it is therefore sufficient to consider the effects in $\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}$.

Lemma 26. *Let $\mathcal{D}$ be an acyclic C^2-BAT, w a world with $w \models \mathcal{D}$, $z \in \mathcal{A}^*$ an action sequence and E_z the effects generated by executing z in w. There exists $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ such that $\mathsf{E}_z \equiv \mathsf{E}'$.*

Proof. Let $\mathsf{fd}(\mathcal{D}) = f$. We prove the claim by induction on the length n of z.

$n = 1$: Let $z = t$ for some $t \in \mathcal{A}$. It holds that $\mathsf{E}_z = \mathcal{E}_{\mathcal{D}}(w, \langle\rangle, t)$. Lemma 23 implies $\mathsf{E}_z \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$.

$n - 1 \Rightarrow n$: Let $z = z' \cdot t$ with $t \in \mathcal{A}$, $z' \in \mathcal{A}^*$ of length $n - 1$ and E the effects generated by executing z' in w. Assume that the claim is true for E. We need to show that for each $\langle F^{\pm}, \varphi\rangle \in \mathsf{E} \rhd \mathcal{E}_{\mathcal{D}}(w, z', t)$ we can find $\langle F^{\pm}, \varphi'\rangle \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ such that $\varphi \equiv \varphi'$.

Lemma 23 yields

$$\mathcal{E}_{\mathcal{D}}(w, z', t) \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}. \tag{14}$$

First, let $\langle F^{\pm}, \varphi\rangle \in \mathsf{E} \rhd \mathcal{E}_{\mathcal{D}}(w, z', t)$ with $\mathsf{fd}_{\mathcal{D}}(F) = 0$. We have to distinguish three cases according to the definition of "$\rhd$".

Case 1: There exists $\langle F^{\pm}, \widehat{\varphi} \rangle \in \mathcal{E}_{\mathcal{D}}(w, z', t)$ such that $\varphi = \mathcal{R}[\mathsf{E}, \widehat{\varphi}]$. Since $\mathsf{fd}_{\mathcal{D}}(F) = 0$ the effect descriptor $\widehat{\varphi}$ does not mention any fluents. Consequently, $\varphi = \mathcal{R}[\mathsf{E}, \widehat{\varphi}] = \widehat{\varphi}$ and with (14) it follows that $\langle F^{\pm}, \varphi \rangle \in \mathcal{E}_{\mathcal{D}}(w, z', t) \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$.

Case 2: Next, we assume that $\langle F^{\pm}, \varphi \rangle$ is a positive effect and there exists $\langle F^{+}, \phi \rangle \in \mathsf{E}$ such that

$$\varphi = \phi \wedge \neg\mathcal{R}[\mathsf{E}, \phi'_1] \wedge \cdots \wedge \neg\mathcal{R}[\mathsf{E}, \phi'_m] \text{ for some } m \geq 0$$

and $\langle F^{-}, \phi'_k \rangle \in \mathcal{E}_{\mathcal{D}}(w, z', t)$ for each $k = 1, \ldots, m$. Since $F(\vec{x})$ has depth 0 we can assume w.l.o.g. that the effect descriptors in $\mathsf{eff}^{+}_{\mathcal{A}}(F) \cup \mathsf{eff}^{-}_{\mathcal{A}}(F)$ are of the form *true* or *false* (if F has arity 0) or $\vec{x} = \vec{c}$ where $\vec{c}$ is an object tuple (if F has arity > 0). By definition of the regression operator we obtain

$$\varphi = \phi \wedge \neg\phi'_1 \wedge \cdots \wedge \neg\phi'_m.$$

Using the assumption $\langle F^{+}, \phi \rangle \in \mathsf{E}$ and the induction hypothesis it follows that ϕ is equivalent to a formula in $\mathsf{eff}^{+}_{\mathcal{A}}(F) \cup \mathsf{eff}^{-}_{\mathcal{A}}(F)$. In case F has arity 0 it follows that φ is equivalent to *true* or *false*. Obviously, it holds that $true \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}_0$ or $false \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}_0$, respectively. In case F has arity > 0 it follows that $\phi \supset \neg\phi'_1 \wedge \cdots \wedge \neg\phi'_m$ due to the standard name assumption. Consequently, $\varphi \equiv \phi$ and the claim holds.

Case 3: $\langle F^{\pm}, \varphi \rangle \in \mathsf{E}$ and $\langle F^{\pm}, \varphi \rangle \in \mathsf{E}$ is a negative effect. The claim follows directly from the induction hypothesis.

Now, let $\langle F^{\pm}, \varphi \rangle \in \mathsf{E} \rhd \mathcal{E}_{\mathcal{D}}(w, z', t)$ with $\mathsf{fd}_{\mathcal{D}}(F) = i$ and $0 < i \leq f$.

Case 1: There exists $\langle F^{\pm}, \widehat{\varphi} \rangle \in \mathcal{E}_{\mathcal{D}}(w, z', t)$ such that $\varphi = \mathcal{R}[\mathsf{E}, \widehat{\varphi}]$. Let $\mathsf{fd}_{\mathcal{D}}(\widehat{\varphi}) = j$. It follows that $j \leq i - 1$. Using Lemma 24 it follows that $\varphi = \mathcal{R}[\mathsf{E}, \widehat{\varphi}] = \mathcal{R}[\mathsf{E}_{\leq j}, \widehat{\varphi}]$. Using the induction hypothesis it follows that there exists a set $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}$ such that $\mathsf{E}_{\leq j} \equiv \mathsf{E}'$. We obtain

$$\varphi = \mathcal{R}[\mathsf{E}, \widehat{\varphi}] = \mathcal{R}[\mathsf{E}_{\leq j}, \widehat{\varphi}] \equiv \mathcal{R}[\mathsf{E}', \widehat{\varphi}].$$

With $\widehat{\varphi} \in \mathsf{eff}^{+}_{\mathcal{A}}(F) \cup \mathsf{eff}^{-}_{\mathcal{A}}(F)$ we obtain $\langle F^{\pm}, \mathcal{R}[\mathsf{E}', \widehat{\varphi}] \rangle \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}_i \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ according to the definition of $\mathfrak{E}^{\mathcal{D},\mathcal{A}}_i$.

Case 2: Next, we assume that $\langle F^{\pm}, \varphi \rangle$ is a positive effect and there exists $\langle F^{+}, \phi \rangle \in \mathsf{E}$ such that

$$\varphi = \phi \wedge \neg\mathcal{R}[\mathsf{E}, \phi'_1] \wedge \cdots \wedge \neg\mathcal{R}[\mathsf{E}, \phi'_m] \text{ for some } m \geq 0$$

and for each $k \in \{1, \ldots, m\}$ there exists $\langle F^{-}, \phi'_k \rangle \in \mathcal{E}_{\mathcal{D}}(w, z', t)$. Since $\mathcal{D}$ is acyclic it holds that $\mathsf{fd}_{\mathcal{D}}(F) > \mathsf{fd}_{\mathcal{D}}(\phi'_k)$ for all k. Therefore we can restrict E to effects on fluents of depth $\leq i - 1$. With Lemma 24 it follows that $\mathcal{R}[\mathsf{E}, \phi'_k] = \mathcal{R}[\mathsf{E}_{\leq i-1}, \phi'_k]$ for all $k \in \{1, \ldots, m\}$. By induction there exists $\widehat{\mathsf{E}} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}$ with $\widehat{\mathsf{E}} \equiv \mathsf{E}_{\leq i-1}$. Consequently using Lemma 25,

$$\varphi \equiv \phi \wedge \neg\mathcal{R}[\widehat{\mathsf{E}}, \phi'_1] \wedge \cdots \wedge \neg\mathcal{R}[\widehat{\mathsf{E}}, \phi'_m] \tag{15}$$

with

$$\{(\phi'_1, \widehat{\mathsf{E}}), \ldots, (\phi'_m, \widehat{\mathsf{E}})\} \subseteq \left[\mathsf{eff}^{-}_{\mathcal{A}}(F) \times 2^{\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}}\right]. \tag{16}$$

Since $\langle F^{+}, \phi \rangle \in \mathsf{E}$ and due to the induction hypothesis for E there are $\zeta \in \mathsf{eff}^{+}_{\mathcal{A}}(F)$, $\mathsf{L} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}$ and $X' \subseteq \left[\mathsf{eff}^{-}_{\mathcal{A}}(F) \times 2^{\mathfrak{E}^{\mathcal{D},\mathcal{A}}_{i-1}}\right]$ such that

$$\phi \equiv \mathcal{R}[\mathsf{L}, \zeta] \wedge \bigwedge_{(\zeta', \mathsf{L}') \in X'} \neg\mathcal{R}[\mathsf{L}', \zeta'] \tag{17}$$

Let

$$X = \{(\phi'_1, \widehat{\mathsf{E}}), \ldots, (\phi'_m, \widehat{\mathsf{E}})\} \cup X'.$$

(15) and (17) yields

$$\varphi \equiv \mathcal{R}[\mathsf{L}, \zeta] \wedge \bigwedge_{(\widehat{\phi}, \mathsf{E}') \in X} \neg\mathcal{R}[\mathsf{E}', \widehat{\phi}]$$

It follows that $\langle F^+, \mathcal{R}[\mathsf{L}, \zeta] \wedge \bigwedge_{(\widehat{\phi}, \mathsf{E}') \in X} \neg\mathcal{R}[\mathsf{E}', \widehat{\phi}]\rangle \in \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ by definition of $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$.

Case 3: $\langle F^\pm, \varphi\rangle \in \mathsf{E}$ is a negative effect. The claim follows directly from the induction hypothesis.

□

Finite abstractions

Using the finite representation of action effects we follow basically the same steps as in [ZC14] to construct finite abstractions of the transition systems generated by executing the program in worlds satisfying an acyclic C^2-BAT. First, we identify a finite set of *relevant C^2-fluent sentences* called context of a program.

Definition 27 (context of a program)**.** Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a Golog program with $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}$, $\mathcal{A}$ the finite set of ground actions and $\mathcal{F}$ the finite set of fluents occurring in $\mathcal{G}$. The *context* $\mathcal{C}(\mathcal{G})$ *of* $\mathcal{G}$ is defined as the smallest set satisfying the following conditions:

- $\mathcal{D}_0 \subseteq \mathcal{C}(\mathcal{G})$
- If $F \in \mathcal{F}$, $t \in \mathcal{A}$ and $(\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \left(\gamma_F^+\right)_t^a \cup \left(\gamma_F^-\right)_t^a$, then $\phi^{\mathsf{con}} \in \mathcal{C}(\mathcal{G})$.
- If ψ? is a test occurring in δ, then $\psi \in \mathcal{C}(\mathcal{G})$.
- If $\psi \in \mathcal{C}(\mathcal{G})$, then $\neg\psi \in \mathcal{C}(\mathcal{G})$ (modulo elimination of double negation).

▲

As in [ZC14] the central notion for the abstraction is that of a *type of a world* representing an equivalence class of worlds. Intuitively, the type of a world tells us which of the context axioms are satisfied in the initial situation and in all relevant future situations of the world.

Definition 28 (type of a world)**.** Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a Golog program with an acyclic BAT $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}$ w.r.t. $\mathcal{A}$, where $\mathcal{A}$ is a finite set of ground actions that includes all ground actions occurring in δ and the two special actions ϵ (for termination) and $\mathfrak{f}$ (for failure). Furthermore, let $\mathcal{C}(\mathcal{G})$ be the context of $\mathcal{G}$ and $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$ the set of all relevant effects according to Definition 22. The *set of all type elements* is given by

$$\mathsf{TE}(\mathcal{G}) := \{(\psi, \mathsf{E}) \mid \psi \in \mathcal{C}(\mathcal{G}), \mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}\}.$$

A *type w.r.t.* $\mathcal{G}$ is a set $\tau \subseteq \mathsf{TE}(\mathcal{G})$ satisfying the following conditions

1. For all $\psi \in \mathcal{C}(\mathcal{G})$ and all $\mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ it holds that either $(\psi, \mathsf{E}) \in \tau$ or $(\neg\psi, \mathsf{E}) \in \tau$.
2. There exists a world $w \in \mathcal{W}$ such that $w \models \mathcal{D}_0 \cup \{\mathcal{R}[\mathsf{E}, \psi] \mid (\psi, \mathsf{E}) \in \tau\}$.

The *set of all types w.r.t.* $\mathcal{G}$ is denoted by $\mathsf{Types}(\mathcal{G})$. Let $w \in \mathcal{W}$ be a world. The *type of* w *w.r.t.* $\mathcal{G}$ is given by

$$\mathsf{type}(w) := \{(\psi, \mathsf{E}) \in \mathsf{TE}(\mathcal{G}) \mid w \models \mathcal{R}[\mathsf{E}, \psi]\}.$$

▲

We show that $\mathsf{Types}(\mathcal{G})$ captures exactly the types of all worlds satisfying the BAT $\mathcal{D}$.

Lemma 29. *Let* $\mathcal{G} = (\mathcal{D}, \delta)$ *and* $\mathsf{Types}(\mathcal{G})$ *be as defined above.*

1. *Let* $w \in \mathcal{W}$ *with* $w \models \mathcal{D}$*. It holds that* $\mathsf{type}(w) \in \mathsf{Types}(\mathcal{G})$*.*
2. *For each* $\tau \in \mathsf{Types}(\mathcal{G})$ *there exists a world* $w \in \mathcal{W}$ *with* $w \models \mathcal{D}$ *such that* $\tau = \mathsf{type}(w)$*.*

Proof.

1. It holds that
   - for all $\psi \in \mathcal{C}(\mathcal{G})$ and all $\mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ either $(\psi, \mathsf{E}) \in \mathsf{type}(w)$ or $(\neg\psi, \mathsf{E}) \in \mathsf{type}(w)$ and
   - $w \models \mathcal{D}$ and $w \models \{\mathcal{R}[\mathsf{E}, \psi] \mid (\psi, \mathsf{E}) \in \mathsf{type}(w)\}$.

   Therefore, $\mathsf{type}(w) \in \mathsf{Types}(\mathcal{G})$.

2. Let $\tau \in \mathsf{Types}(\mathcal{G})$. There exists a world w such that $w \models \mathcal{D}_0 \cup \{\mathcal{R}[\mathsf{E}, \psi] \mid (\psi, \mathsf{E}) \in \tau\}$. It follows that $\tau = \mathsf{type}(w)$. We define a world $w_{\mathcal{D}}$ with $w_{\mathcal{D}} \models \mathcal{D}$ satisfying $\mathsf{type}(w) = \mathsf{type}(w_D)$. Let $\mathcal{F}$ be the set of fluents occurring in $\mathcal{G}$ with an SSA in $\mathcal{D}_{\text{post}}$. Given the world w, we define $w_{\mathcal{D}}$ as the world satisfying the following conditions:
   - For all $F(\vec{n}) \in \mathcal{P}_F$ with $F \notin \mathcal{F}$ and all $z \in \mathcal{Z}$: $w_{\mathcal{D}}[F(\vec{n}), z] = w[F(\vec{n}), z]$.
   - For all $F(\vec{n}) \in \mathcal{P}_F$ with $F \in \mathcal{F}$:
     - $w_{\mathcal{D}}[F(\vec{n}), \langle\rangle] = w[F(\vec{n}), \langle\rangle]$ and
     - $w_{\mathcal{D}}[F(\vec{n}), z \cdot t] = 1$ iff $w_{\mathcal{D}}, z \models \big(\gamma_F^+\big)^{a\,\vec{x}}_{t\,\vec{n}} \lor F(\vec{n}) \land \neg\big(\gamma_F^-\big)^{a\,\vec{x}}_{t\,\vec{n}}$ for all $z \cdot t \in \mathcal{Z}$.

   It is easy to see that $w_{\mathcal{D}} \in \mathcal{W}$ exists, is uniquely determined and satisfies $w_{\mathcal{D}} \models \mathcal{D}$. Using Lemma 21 it follows that $\mathsf{type}(w) = \mathsf{type}(w_{\mathcal{D}})$.

□

The abstraction of a situation consisting of a world $w \in \mathcal{W}$ with $w \models \mathcal{D}$ and an action sequence $z \in \mathcal{A}^*$ is then given by $\mathsf{type}(w)$ and the set of effects E_z generated by executing z in w. We define an abstract version of the effect function (see Definition 16).

Definition 30. Let $\mathcal{G} = (\mathcal{D}, \delta)$, $\mathfrak{E}^{\mathcal{D},\mathcal{A}}$ and $\mathsf{Types}(\mathcal{G})$ be as above. Let $\tau \in \mathsf{Types}(\mathcal{G})$, $\mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ and $t \in \mathcal{A}$. The effects of executing t in (τ, E) are given by

$$\begin{aligned}\widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, t) :=&\{\langle F^+, \phi^{\mathsf{eff}}\rangle \mid \exists(\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \big(\gamma_F^+\big)^{a}_{t} \text{ s.t. } (\phi^{\mathsf{con}}, \mathsf{E}) \in \tau\} \cup \\ &\{\langle F^-, \phi^{\mathsf{eff}}\rangle \mid \exists(\phi^{\mathsf{eff}}, \phi^{\mathsf{con}}) \in \big(\gamma_F^-\big)^{a}_{t} \text{ s.t. } (\phi^{\mathsf{con}}, \mathsf{E}) \in \tau\}.\end{aligned}$$

▲

We show that the concrete effect function and the abstract one yield the same result.

Lemma 31. *Let w be a world with $w \models \mathcal{D}$, $z \in \mathcal{A}^*$, $t \in \mathcal{A}$ and E_z the effects generated by executing z in w. For $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ with $\mathsf{E}_z \equiv \mathsf{E}'$ it holds that $\mathcal{E}_{\mathcal{D}}(w, z, t) = \widehat{\mathcal{E}}_{\mathcal{D}}(\mathsf{type}(w), \mathsf{E}', t)$.*

Proof. Let w, z, t and E_z be as stated in the claim. Lemma 26 implies that there exists a set of effects $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ with $\mathsf{E}_z \equiv \mathsf{E}'$. Lemma 29 yields $\mathsf{type}(w) \in \mathsf{Types}(\mathcal{G})$. Therefore, $\widehat{\mathcal{E}}_{\mathcal{D}}(\mathsf{type}(w), \mathsf{E}', t)$ is well defined. It holds that $\langle F^{\pm}, \phi\rangle \in \mathcal{E}_{\mathcal{D}}(w, z, t)$

iff there exists $(\phi, \psi) \in \left(\gamma_F^+\right)_t^a \cup \left(\gamma_F^-\right)_t^a$ such that $w, z \models \psi$

iff there exists $(\phi, \psi) \in \left(\gamma_F^+\right)_t^a \cup \left(\gamma_F^-\right)_t^a$ such that $w \models \mathcal{R}[\mathsf{E}_z, \psi]$ (by Lemma 21)

iff there exists $(\phi, \psi) \in \left(\gamma_F^+\right)_t^a \cup \left(\gamma_F^-\right)_t^a$ such that $w \models \mathcal{R}[\mathsf{E}', \psi]$ (with $\mathsf{E}_z \equiv \mathsf{E}'$)

iff there exists $(\phi, \psi) \in \left(\gamma_F^+\right)_t^a \cup \left(\gamma_F^-\right)_t^a$ such that $(\psi, \mathsf{E}') \in \mathsf{type}(w)$ (by Def. 27 and 28)

iff $\langle F^{\pm}, \phi\rangle \in \widehat{\mathcal{E}}_{\mathcal{D}}(\mathsf{type}(w), \mathsf{E}', t)$.

□

Next, we introduce additional notions for traversing the space of reachable subprograms. Before executing the next action according to a program we first need to perform the necessary tests.

Definition 32 (guarded action)**.** Let δ be a program expression over ground actions $\mathcal{A} \subset \mathcal{N}_A$ (including the termination action ϵ). A guarded action in δ is of the form $\psi_1?; \cdots; \psi_n?; t$ for some $n \geq 0$ where $t \in Act$ and $\psi_1, \ldots, \psi_n$ are tests occurring in δ. ▲

We use the symbol $\mathfrak{a}$ to denote a guarded action. Analogous to [BZ13, ZC13] we define two functions $\mathsf{head}(\cdot)$ and $\mathsf{tail}(\cdot,\cdot)$. Intuitively, $\mathsf{head}(\delta)$ contains those guarded actions that can be executed first when executing δ and $\mathsf{tail}(\mathfrak{a}, \delta)$ yields the program expressions that remain to be executed after executing the guarded action $\mathfrak{a}$ from the head of δ.

Definition 33. The function $\mathsf{head}(\cdot)$ maps a program expression to a set of guarded actions in this program expression. It is defined by induction on the structure of program expressions:

1. $\mathsf{head}(\langle\rangle) := \{\epsilon\}$;
2. $\mathsf{head}(t) := \{t\}$ for all $t \in \mathcal{A}$;
3. $\mathsf{head}(\psi?) := \{\psi?; \epsilon\}$;
4. $\mathsf{head}(\delta^*) := \{\epsilon\} \cup \mathsf{head}(\delta)$;
5. $\mathsf{head}(\delta_1; \delta_2) := \{\mathfrak{a} \mid \mathfrak{a} = \psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\delta_1) \wedge t \neq \epsilon\} \cup$
$\{\psi_1?; \cdots; \psi_n?; \psi_1'?; \cdots; \psi_m'?; t \mid \psi_1?; \cdots; \psi_n?; \epsilon \in \mathsf{head}(\delta_1) \wedge$
$\psi_1'?; \cdots; \psi_m'?; t \in \mathsf{head}(\delta_2)\}$;
6. $\mathsf{head}(\delta_1|\delta_2) := \mathsf{head}(\delta_1) \cup \mathsf{head}(\delta_2)$;
7. $\mathsf{head}(\delta_1\|\delta_2) := \{\mathfrak{a} \mid \mathfrak{a} = \psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\delta_i) \wedge i \wedge t \neq \epsilon\} \cup$
$\{\psi_1?; \cdots; \psi_n?; \psi_1'?; \cdots; \psi_m'?; t \mid \psi_1?; \cdots; \psi_n?; \epsilon \in \mathsf{head}(\delta_i) \wedge$
$\psi_1'?; \cdots; \psi_m'?; t \in \mathsf{head}(\delta_j) \wedge$
$\{i, j\} = \{1, 2\}\}$.

▲

Definition 34. The function $\mathsf{tail}(\cdot,\cdot)$ maps a guarded action and a program expression to a set of program expressions.

- If $\mathfrak{a} \notin \mathsf{head}(\delta)$, then $\mathsf{tail}(\mathfrak{a},\delta) = \emptyset$.
- If $\mathfrak{a} \in \mathsf{head}(\delta)$ and $\mathfrak{a} = \psi_1?; \cdots ; \psi_n?; \epsilon$, then $\mathsf{tail}(\mathfrak{a},\delta) = \{\langle\rangle\}$.
- If $\mathfrak{a} \in \mathsf{head}(\delta)$ and $\mathfrak{a} = \psi_1?; \cdots ; \psi_n?; t$ for $t \in \mathcal{A}\setminus\{\epsilon\}$, then $\mathsf{tail}(\mathfrak{a},\delta)$ is defined by induction on the combined size of $\mathfrak{a}$ and δ:
  1. $\mathsf{tail}(\mathfrak{a},t') := \{\langle\rangle\}$ for $t \in \mathcal{A}$;[1]
  2. $\mathsf{tail}(\mathfrak{a},\delta^*) := \{\delta';(\delta)^* \mid \delta' \in \mathsf{tail}(\mathfrak{a},\delta)\}$;
  3. $$\begin{aligned}\mathsf{tail}(\mathfrak{a},\delta_1;\delta_2) := \; & \{\delta';\delta_2 \mid \delta' \in \mathsf{tail}(\mathfrak{a},\delta_1)\} \cup \\ & \{\delta'' \mid \exists 0 \leq i \leq n \text{ s.t. } \mathfrak{a} = \psi_1; \cdots ; \psi_i?; \cdots ; \psi_n?; t \, \wedge \\ & \qquad \psi_1?; \cdots ; \psi_i?; \epsilon \in \mathsf{head}(\delta_1) \, \wedge \\ & \qquad \delta'' \in \mathsf{tail}(\psi_{i+1}?; \cdots ; \psi_n?; t, \delta_2)\};\end{aligned}$$
  4. $\mathsf{tail}(\mathfrak{a},\delta_1|\delta_2) := \mathsf{tail}(\mathfrak{a},\delta_1) \cup \mathsf{tail}(\mathfrak{a},\delta_2)$.
  5. $$\begin{aligned}\mathsf{tail}(\mathfrak{a},\delta_1\|\delta_2) := \; & \{\delta'\|\delta_2 \mid \delta' \in \mathsf{tail}(\mathfrak{a},\delta_1)\} \cup \{\delta_1\|\delta' \mid \delta' \in \mathsf{tail}(\mathfrak{a},\delta_2)\} \cup \\ & \{\delta'' \mid \psi_1?; \cdots ; \psi_n?; \epsilon \in \mathsf{head}(\delta_i) \wedge \psi'_1?; \cdots ; \psi'_m?; t \in \mathsf{head}(\delta_j) \, \wedge \\ & \qquad \delta'' \in \mathsf{tail}(\psi'_1?; \cdots ; \psi'_m?; t, \delta_j) \wedge \{i,j\} = \{1,2\} \, \wedge \\ & \qquad \mathfrak{a} \text{ is of the form } \psi_1?; \cdots ; \psi_n?; \psi'_1?; \cdots ; \psi'_m?; t\}.\end{aligned}$$

▲

We establish the relationship of the head and tail functions with the transition semantics.

Lemma 35. *Let $\mathcal{G} = (\mathcal{D},\delta)$ be a program, w a world with $w \models \mathcal{D}$ and $\langle z,\rho\rangle \in \mathsf{Reach}(w,\delta)$. It holds that*

1. *$\langle z,\rho\rangle \in \mathsf{Fin}(w)$ iff there exists $\psi_1?; \cdots ; \psi_n?; \epsilon \in \mathsf{head}(\rho)$ and $w, z \models \psi_i$ for all $i = 1, \ldots, n$;*
2. *$\langle z,\rho\rangle \xrightarrow{w} \langle z \cdot t, \rho'\rangle$ iff there exists $\mathfrak{a} = \psi_1?; \cdots ; \psi_n?; t \in \mathsf{head}(\rho)$ with $t \neq \epsilon$, $w, z \models \psi_i$ for all $i = 1, \ldots, n$ and $\rho' \in \mathsf{tail}(\mathfrak{a},\rho)$;*
3. *$\langle z,\rho\rangle \in \mathsf{Fail}(w)$ iff there exists* no *$\mathfrak{a} = \psi_1?; \cdots ; \psi_n?; t \in \mathsf{head}(\rho)$ such that $w, z \models \psi_i$ for all $i = 1, \ldots, n$.*

Proof. The claims can be shown by induction on the structure of ρ. The proof is analogous to the proof of Lemma 15, page 9 in [ZC13]. □

We define the set of reachable subprograms using the head and tail functions.

Definition 36. 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 $i = 0, \ldots, n-1$ 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 $\mathsf{Sub}(\delta)$. ▲

As shown in [BZ13] the set $\mathsf{Sub}(\delta)$ is finite. Now we are ready to define the finite propositional abstraction of a transition system.

[0] Note that $\mathfrak{a} \in \mathsf{head}(t')$ implies $\mathfrak{a} = t'$.

Definition 37. Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a program with an acyclic C^2-BAT $\mathcal{D}$ w.r.t. the ground actions occurring in δ and let $\mathsf{Types}(\mathcal{G})$ and $\mathcal{C}(\mathcal{G})$ be as defined above. We define a set of atomic propositions by introducing for each axiom ψ in $\mathcal{C}(\mathcal{G})$ a corresponding atomic proposition p_ψ:

$$\mathsf{P}_{\mathcal{C}(\mathcal{G})} := \{\mathsf{p}_\psi \mid \psi \in \mathcal{C}(\mathcal{G})\}.$$

Let $\tau \in \mathsf{Types}(\mathcal{G})$. The corresponding *propositional transition system* $\mathcal{T}^\tau_\delta = \big(S_{\tau,\delta}, \overset{\tau,\delta}{\Longrightarrow}, L_\mathcal{C}\big)$ consists of a set of states given by

$$S_{\tau,\delta} := 2^{\mathfrak{E}^{\mathcal{D},\mathcal{A}}} \times \mathsf{Sub}(\delta),$$

a transition relation $\overset{\tau,\delta}{\Longrightarrow} \subseteq S_{\tau,\delta} \times S_{\tau,\delta}$ with $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \rho')$ iff

- there exists $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that
  - $\{(\psi_1, \mathsf{E}), \ldots, (\psi_n, \mathsf{E})\} \subseteq \tau$,
  - $\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_\mathcal{D}(\tau, \mathsf{E}, t)$ and
  - $\rho' \in \mathsf{tail}(\psi_1?; \cdots; \psi_n?; t, \rho)$

  or
- there exists *no* $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $\{(\psi_1, \mathsf{E}), \ldots, (\psi_n, \mathsf{E})\} \subseteq \tau$ and $\rho = \rho'$ and $\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_\mathcal{D}(\tau, \mathsf{E}, \mathfrak{f})$

and a labeling function $L_\mathcal{C}$ such that

$$L_\mathcal{C} : (\mathsf{E}, \rho) \mapsto \{\mathsf{p}_\psi \in \mathsf{P}_{\mathcal{C}(\mathcal{G})} \mid (\psi, \mathsf{E}) \in \tau\}$$

for all $(\mathsf{E}, \rho) \in S_{\tau,\delta}$. ▲

Consider a program $\mathcal{G} = (\mathcal{D}, \delta)$ and a world $w \models \mathcal{D}$. We define a relation

$$\simeq_{w,\delta} \subseteq \mathsf{Reach}(w, \delta) \times S_{\tau,\delta} \text{ with } \tau = \mathsf{type}(w)$$

satisfying the following condition: It holds that $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho')$ iff $\mathsf{E} \equiv \mathsf{E}_z$, where E_z are the effects generated by executing z in w, and $\rho = \rho'$.

Lemma 38. *Let $\mathcal{G} = (\mathcal{D}, \delta)$ be as above, $w \in \mathcal{W}$ with $w \models \mathcal{D}$ and $\tau := \mathsf{type}(w)$ and let $\langle z, \rho \rangle \in \mathsf{Reach}(w, \delta)$ and $(\mathsf{E}, \rho) \in S_{\tau,\delta}$ such that $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$.*

1. *For all $\psi \in \mathcal{C}(\mathcal{G})$ it holds that $w, z \models \psi$ iff $\mathsf{p}_\psi \in L_\mathcal{C}(\mathsf{E}, \rho)$.*
2. *For all $\langle z', \rho' \rangle \in \mathsf{Reach}(w, \delta)$ with $\langle z, \rho \rangle \overset{w,\delta}{\hookrightarrow} \langle z', \rho' \rangle$ there exists $(\mathsf{E}', \rho') \in S_{\tau,\delta}$ such that $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \rho')$ and $\langle z', \rho' \rangle \simeq_{w,\delta} (\mathsf{E}', \rho')$.*
3. *For all $(\mathsf{E}', \rho') \in S_{\tau,\delta}$ with $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \rho')$ there exists $\langle z', \rho' \rangle \in \mathsf{Reach}(w, \delta)$ such that $\langle z, \rho \rangle \overset{w,\delta}{\hookrightarrow} \langle z', \rho' \rangle$ and $\langle z', \rho' \rangle \simeq_{w,\delta} (\mathsf{E}', \rho')$.*

Proof.

1. Let $\psi \in \mathcal{C}(\mathcal{G})$ and let E_z be the effects generated by executing z in w. By definition $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ implies $\mathsf{E}_z \equiv \mathsf{E}$. It holds that $w, z \models \psi$

iff $w \models \mathcal{R}[\mathsf{E}_z, \psi] \equiv \mathcal{R}[\mathsf{E}, \psi]$ (by Lemma 21)
iff $(\psi, \mathsf{E}) \in \tau$ (by definition of types)
iff $\mathsf{p}_\psi \in L_{\mathcal{C}}(\mathsf{E}, \rho)$.

2. Let $\langle z \cdot t, \rho' \rangle \in \mathsf{Reach}(w, \delta)$ with $\langle z, \rho \rangle \overset{w,\delta}{\hookrightarrow} \langle z \cdot t, \rho' \rangle$. We distinguish the three cases with $t \notin \{\epsilon, \mathfrak{f}\}$, $t = \epsilon$ or $t = \mathfrak{f}$.

    (a) $t \notin \{\epsilon, \mathfrak{f}\}$ implies $\langle z, \rho \rangle \xrightarrow{w} \langle z \cdot t, \rho' \rangle$. The second claim of Lemma 35 implies that there is $\psi_1?; \cdots, \psi_n?; t \in \mathsf{head}(\rho)$ such that $w, z \models \psi_1 \wedge \cdots \wedge \psi_n$ and $\rho' \in \mathsf{tail}(\psi_1?; \cdots, \psi_n?; t, \rho)$. By definition of the context it holds that $\{\psi_1, \dots, \psi_n\} \subseteq \mathcal{C}(\mathcal{G})$. By assumption it holds that $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ and the first claim of this lemma yields $\{\mathsf{p}_{\psi_1}, \dots, \mathsf{p}_{\psi_n}\} \subseteq L_{\mathcal{C}}(\mathsf{E}, \rho)$ and therefore

    $$\{(\psi_1, \mathsf{E}), \dots, (\psi_n, \mathsf{E})\} \subseteq \tau.$$

    By Lemma 26, $\mathsf{E} \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ implies that there exists a set of effects $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ such that $\mathsf{E}' \equiv \mathsf{E} \rhd \mathcal{E}_{\mathcal{D}}(w, z, t)$. Lemma 31 yields $\mathcal{E}_{\mathcal{D}}(w, z, t) = \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, t)$ and $\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, t)$. By definition of $\overset{\tau,\delta}{\Longrightarrow}$ we obtain $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \rho')$.
    With Lemma 25 and $\mathsf{E} \equiv \mathsf{E}_z$ it follows that $\mathsf{E}' \equiv \mathsf{E}_z \rhd \mathcal{E}_{\mathcal{D}}(w, z, t)$ and therefore

    $$\langle z \cdot t, \rho' \rangle \simeq_{w,\delta} (\mathsf{E}', \rho').$$

    (b) $t = \epsilon$ implies $\langle z, \rho \rangle \in \mathsf{Fin}(w)$ and $\rho' = \langle\rangle$. The first claim of Lemma 35 implies that $\psi_1?; \cdots; \psi_n?; \epsilon \in \mathsf{head}(\rho)$ and $w, z \models \psi_1 \wedge \cdots \wedge \psi_n$. Since $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ and $\{\psi_1, \dots, \psi_n\} \subseteq \mathcal{C}(\mathcal{G})$ it follows that $\{(\psi_1, \mathsf{E}), \dots, (\psi_n, \mathsf{E})\} \subseteq \tau$. There exists $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ with

    $$\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, \epsilon) \equiv \mathsf{E}_z \rhd \mathcal{E}_{\mathcal{D}}(w, z, \epsilon).$$

    Consequently, there is a transition $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \langle\rangle)$ with $\langle z \cdot \epsilon, \langle\rangle \rangle \simeq_{w,\delta} (\mathsf{E}', \langle\rangle)$.

    (c) $t = \mathfrak{f}$ implies that $\langle z, \rho \rangle \in \mathsf{Fail}(w)$ and $\rho = \rho'$. The third claim of Lemma 35 implies that there is *no* $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $w, z \models \psi_1 \wedge \cdots \wedge \psi_n$. For all $\varphi_1?; \cdots; \varphi_n?; t \in \mathsf{head}(\rho)$ it holds that $\{\varphi_1, \dots, \varphi_n\} \subseteq \mathcal{C}(\mathcal{G})$. Since $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ it follows that there is *no* $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $\{(\psi_1, \mathsf{E}), \dots, (\psi_n, \mathsf{E})\} \subseteq \tau$. There exists $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ with

    $$\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, \mathfrak{f}) \equiv \mathsf{E}_z \rhd \mathcal{E}_{\mathcal{D}}(w, z, \mathfrak{f}).$$

    Consequently, there is a transition $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \langle\rangle)$ with $\langle z \cdot \mathfrak{f}, \langle\rangle \rangle \simeq_{w,\delta} (\mathsf{E}', \langle\rangle)$.

3. Let $(\mathsf{E}', \rho') \in S_{\tau,\delta}$ such that $(\mathsf{E}, \rho) \overset{\tau,\delta}{\Longrightarrow} (\mathsf{E}', \rho')$.

    (a) There exists $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $\{(\psi_1, \mathsf{E}), \dots, (\psi_n, \mathsf{E})\} \subseteq \tau$, $\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, t)$ and $\rho' \in \mathsf{tail}(\psi_1?; \cdots; \psi_n?; t, \rho)$. Since $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ it follows that $w, z \models \psi_1 \wedge \cdots \wedge \psi_n$.

        i. Assume $t \neq \epsilon$. Lemma 35 implies that $\langle z, \rho \rangle \xrightarrow{w} \langle z \cdot t, \rho' \rangle$. Lemma 31 implies $\langle z \cdot t, \rho \rangle \simeq_{w,\delta} (\mathsf{E}', \rho')$.

        ii. Assume $t = \epsilon$. We have $\rho' = \langle\rangle$. Lemma 35 implies that $\langle z, \rho \rangle \in \mathsf{Fin}(w)$. Therefore, there is a transition $\langle z, \rho \rangle \overset{w,\delta}{\hookrightarrow} \langle z \cdot \epsilon, \langle\rangle \rangle$. As in the previous case it follows that $\langle z \cdot \epsilon, \rho \rangle \simeq_{w,\delta} (\mathsf{E}', \rho')$.

    (b) There exists *no* $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $\{(\psi_1, \mathsf{E}), \dots, (\psi_n, \mathsf{E})\} \subseteq \tau$ and $\rho = \rho'$ and $\mathsf{E}' \equiv \mathsf{E} \rhd \widehat{\mathcal{E}}_{\mathcal{D}}(\tau, \mathsf{E}, \mathfrak{f})$. Since $\langle z, \rho \rangle \simeq_{w,\delta} (\mathsf{E}, \rho)$ it follows that there is no $\psi_1?; \cdots; \psi_n?; t \in \mathsf{head}(\rho)$ such that $w, z \models \psi_1 \wedge \cdots \wedge \psi_n$. Lemma 35 yields $\langle z, \rho \rangle \in \mathsf{Fail}(w)$. There is a transition $\langle z, \rho \rangle \overset{w,\delta}{\hookrightarrow} \langle z \cdot \mathfrak{f}, \rho \rangle$. With Lemma 31 it follows that $\langle z \cdot \mathfrak{f}, \rho \rangle \simeq_{w,\delta} (\mathsf{E}', \rho)$.

□

Consider a temporal state formula Φ and a temporal path formula Ψ over axioms from $\mathcal{C}(\mathcal{G})$. From Φ and Ψ we obtain a propositional CTL* state formula and CTL* path formula, respectively, by replacing each axiom ψ in Φ and Ψ, respectively, by the corresponding proposition $\mathsf{p}_\psi \in \mathsf{P}_{\mathcal{C}(\mathcal{G})}$. The resulting formulas are denoted by $\mathsf{pr}(\Phi)$ and $\mathsf{pr}(\Psi)$, respectively. Given a state s in the propositional transition system $\mathcal{T}_\delta^\tau = \big(S_{\tau,\delta}, \xRightarrow{\tau,\delta}, L_\mathcal{C}\big)$, satisfaction of $\mathsf{pr}(\Phi)$ in $\mathcal{T}_\delta^\tau, s$ denoted by $\mathcal{T}_\delta^\tau, s \models \mathsf{pr}(\Phi)$ is defined in the standard way [BK08]. Similarly, for an infinite path π in $\mathcal{T}_\delta^\tau = \big(S_{\tau,\delta}, \xRightarrow{\tau,\delta}, L_\mathcal{C}\big)$ satisfaction of $\mathsf{pr}(\Psi)$ in $\mathcal{T}_\delta^\tau, \pi$ denoted by $\mathcal{T}_\delta^\tau, \pi \models \mathsf{pr}(\Psi)$ is defined accordingly [BK08].

Lemma 39. *Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a program satisfying the acyclicity condition, Φ a temporal state formula over axioms in $\mathcal{C}(\mathcal{G})$ and w a world with $w \models \mathcal{D}$. It holds that*

$$\mathsf{T}_\delta^w, \langle\langle\rangle, \delta\rangle \models \Phi \textit{ iff } \mathcal{T}_\delta^{\mathsf{type}(w)}, (\emptyset, \delta) \models \mathsf{pr}(\Phi).$$

Proof. This lemma is a consequence of Lemma 38. □

To decide whether a temporal state formula Φ over axioms in $\mathcal{C}(\mathcal{G})$ is valid in $\mathcal{G} = (\mathcal{D}, \delta)$ with an acyclic action theory we first compute $\mathsf{Types}(\mathcal{G})$. For each $\tau \in \mathsf{Types}(\mathcal{G})$ the finite propositional transition system $\mathcal{T}_\delta^\tau$ can be computed. Finally, we check for each $\tau \in \mathsf{Types}(\mathcal{G})$ whether $\mathcal{T}_\delta^\tau, (\emptyset, \delta) \models \mathsf{pr}(\Phi)$ holds using model checking.

Theorem 40. *Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a program with a C^2-BAT that is acyclic and Φ a temporal state formula over axioms in $\mathcal{C}(\mathcal{G})$. It is decidable to verify whether Φ is valid in $\mathcal{G}$.*

3.3 Decidable Verification with Flat Action Theories

The techniques introduced for acyclic theories can also be applied to programs with a C^2-BAT $\mathcal{D}$ where all the effect descriptors in the SSAs in $\mathcal{D}$ are quantifier-free but may contain cycles. (The domain in Example 6 satisfies also this restriction). We call this class *flat action theory.* It is straightforward to show that in this case only finitely many effects can be generated. We use the same arguments as for the acyclic case to show that a finite abstraction of the transition system can be constructed such that satisfaction of temporal properties is preserved.

Definition 41. Let $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_{\text{post}}$ be a C^2-BAT. We say that $\mathcal{D}$ is a *flat action theory* if for each effect condition $\gamma_F^\pm$ occurring in $\mathcal{D}_{\text{post}}$ and all disjuncts

$$\exists \vec{y}.\big(a = A(\vec{v}) \wedge \phi \wedge \phi'\big)$$

occurring in $\gamma_F^\pm$ the effect descriptors ϕ are quantifier-free. ▲

Based on the finite set of fluents $\mathcal{F}$ occurring in a program $\mathcal{G} = (\mathcal{D}, \delta)$ with a flat action theory $\mathcal{D}$ and the constants occurring in $\mathcal{G} = (\mathcal{D}, \delta)$ there are finitely many atomic C^2-fluent formulas of the following forms:

- F'', $F'(x)$, $F'(y)$, $F(x,y)$, $F(y,x)$, $F(x,x)$, $F(y,y)$,
- $F'(c)$, $F(c,x)$, $F(c,y)$, $F(x,c)$, $F(y,c)$, $F(c,c')$,
- $x = c$, $y = c$ or $c = c'$,

where $\{F'', F', F\} \subseteq \mathcal{F}$ and c and c' are constants occurring in $\mathcal{D}$. We denote this finite set of atoms by $\mathsf{At}(\mathcal{G})$ and in addition require that $\mathsf{At}(\mathcal{G})$ is closed under negation.

Note that the regression operator $\mathcal{R}[\mathsf{E}, \phi]$ in Figure 5 introduces new quantifiers for the cases $\mathcal{R}[\mathsf{E}, F(x, x)]$ and $\mathcal{R}[\mathsf{E}, F(y, y)]$. Here we only need to consider quantifier-free effects in E and we modify the regression operator for the two cases as follows:

$$\mathcal{R}[\mathsf{E}, F(x,x)] := F(x,x) \wedge \bigwedge_{\langle F(x,y)^-,\varphi\rangle \in \mathsf{E}} \wedge \neg \varphi_x^y \vee \bigvee_{\langle F(x,y)^+,\varphi\rangle \in \mathsf{E}} \varphi_x^y;$$

$$\mathcal{R}[\mathsf{E}, F(y,y)] := F(y,y) \wedge \bigwedge_{\langle F(x,y)^-,\varphi\rangle \in \mathsf{E}} \wedge \neg \varphi_y^x \vee \bigvee_{\langle F(x,y)^+,\varphi\rangle \in \mathsf{E}} \varphi_y^x.$$

Consequently, if ϕ and the effects in E are quantifier-free, then $\mathcal{R}[\mathsf{E}, \phi]$ is quantifier-free as well. A quantifier-free C^2-formula can be equivalently transformed into *conjunctive normal form* (CNF) and can be viewed as a set of sets of atoms in $\mathsf{At}(\mathcal{G})$. For a flat theory $\mathcal{D}$ and the set of ground action $\mathcal{A}$ in $\mathcal{G} = (\mathcal{D}, \delta)$ the effect descriptors in $\mathsf{eff}^+_{\mathcal{A}}(F) \cup \mathsf{eff}^-_{\mathcal{A}}(F)$ are boolean combinations of atoms in $\mathsf{At}(\mathcal{G})$ for all fluents in $\mathcal{F}$. We define the set of relevant effects as follows:

$$\begin{aligned}\mathfrak{E}^{\mathcal{D},\mathcal{A}} := \{\langle F^{\pm}, \varphi\rangle \mid\ & F(\vec{x}) \in \mathcal{F}, \varphi \text{ has free variables } \vec{x}, \\ & \varphi \text{ is in CNF consisting of atoms in } \mathsf{At}(\mathcal{G})\}.\end{aligned}$$

We can now easily prove Lemma 26 for flat action theories as well.

Lemma 42. *Let $\mathcal{D}$ be a flat C^2-BAT, w a world with $w \models \mathcal{D}$, $z \in \mathcal{A}^*$ an action sequence and E_z the effects generated by executing z in w. There exists $\mathsf{E}' \subseteq \mathfrak{E}^{\mathcal{D},\mathcal{A}}$ such that $\mathsf{E}_z \equiv \mathsf{E}'$.*

Using the same abstraction technique as for programs with acyclic theories we obtain our decidability result for the verification problem.

Theorem 43. *Let $\mathcal{G} = (\mathcal{D}, \delta)$ be a program with a flat C^2-BAT and Φ a temporal state formula over axioms in $\mathcal{C}(\mathcal{G})$. It is decidable to verify whether Φ is valid in $\mathcal{G}$.*

4 Related Work

De Giacomo, Lespérance and Patrizi [DLP12] show decidability for first-order μ-calculus properties for a class of BATs where fluent extensions are bounded by some fixed threshold. Moreover, their notion of boundedness is a semantical condition that is in general undecidable to check, whereas our approach relies on purely syntactical restrictions. [HCMD14] investigate acyclicity conditions that ensure *state-boundedness* in data-aware dynamic systems. State-boundedness then in turn allows for decidable verification by constructing finite abstraction of infinite transition systems. However, the setting is quite different: The transition systems in [HCMD14] have a fixed database instance as initial state, actions do not respect the frame assumption but for example may cause an infinite branching degree.

5 Conclusion

In this paper we broadened the class of Golog programs and action theories for which decidability of verification can be achieved. The new class of acyclic theories subsumes many of the ones that were previously studied, including the context-free and local-effect ones and also the class considered in Theorem 43 subsumes local-effect theories. We observe that the decidability does not merely depend on whether actions may affect an unbounded number of objects, i.e. have non-local effects, but also on the dependencies between fluents in the action theory. Interestingly, it turns out that in domains as the one described in Example 6, in the *briefcase domain* [Ped88], or in the *logistics domain* [Bac01], actions have non-local effects but dependencies are acyclic. Note that we refer to non-propositional models of the domains in the Situation Calculus, i.e. ones that admit a (possibly) infinite number of objects.

References

[Bac01] Bacchus, Fahiem: The AIPS '00 Planning Competition. In: *AI Magazine* 22 (2001), Nr. 3, 47–56. `http://www.aaai.org/ojs/index.php/aimagazine/article/view/1571`

[BK08] Baier, Christel ; Katoen, Joost-Pieter: *Principles of model checking.* MIT Press, 2008. – ISBN 978–0–262–02649–9

[BZ13] 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)

[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

[CLLZ14] Classen, Jens ; Liebenberg, Martin ; Lakemeyer, Gerhard ; Zarriess, Benjamin: Exploring the Boundaries of Decidable Verification of Non-Terminating Golog Programs. In: Brodley, Carla E. (Hrsg.) ; Stone, Peter (Hrsg.): *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014)*, AAAI Press, 2014, S. 1012–1019

[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

[GS07] Gu, Yilan ; Soutchanski, Mikhail: Decidable Reasoning in a Modified Situation Calculus. In: Veloso, Manuela M. (Hrsg.): *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, AAAI Press, 2007, S. 1891–1897

[HCMD14] Hariri, Babak B. ; Calvanese, Diego ; Montali, Marco ; Deutsch, Alin: State-Boundedness in Data-Aware Dynamic Systems. In: Eiter, Thomas (Hrsg.) ; Baral, Chitta (Hrsg.) ; Giacomo, Giuseppe D. (Hrsg.): *Proceedings of the Fourtenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2014)*, AAAI Press, 2014

[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

[LR97] Lin, Fangzhen ; Reiter, Raymond: How to Progress a Database. In: *Artificial Intelligence* 92 (1997), Nr. 1–2, S. 131–167

[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

[McI00] McIlraith, Sheila A.: Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes). In: *Artificial Intelligence* 116 (2000), Nr. 1-2, S. 87–121

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

[Ped88] Pednault, Edwin P. D.: Synthesizing plans that contain actions with context-dependent effects. In: *Computational Intelligence* 4 (1988), S. 356–372. `http://dx.doi.org/10.1111/j.1467-8640.1988.tb00285.x`. – DOI 10.1111/j.1467–8640.1988.tb00285.x

[Rei01] Reiter, Raymond: *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001

[VLL08] Vassos, Stavros ; Lakemeyer, Gerhard ; Levesque, Hector J.: First-Order Strong Progression for Local-Effect Basic Action Theories. 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. 662–672

[ZC13] Zarriess, Benjamin ; Classen, Jens: On the Decidability of Verifying LTL Properties of Golog Programs / Chair of Automata Theory, TU Dresden. Dresden, Germany, 2013 (13–10). – LTCS-Report

[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