



# Performability of Actions

Janusz Czelakowski<sup>1</sup>

Accepted: 6 September 2021 / Published online: 6 October 2021  
© The Author(s) 2021

## Abstract

Action theory may be regarded as a theoretical foundation of AI, because it provides in a logically coherent way the principles of performing actions by agents. But, more importantly, action theory offers a formal ontology mainly based on set-theoretic constructs. This ontology isolates various types of actions as structured entities: atomic, sequential, compound, ordered, situational actions etc., and it is a solid and non-removable foundation of any rational activity. The paper is mainly concerned with a bunch of issues centered around the notion of performability of actions. It seems that the problem of performability of actions, though of basic importance for purely practical applications, has not been investigated in the literature in a systematic way thus far. This work, being a companion to the book as reported (Czelakowski in Freedom and enforcement in action. Elements of formal action theory, Springer 2015), elaborates the theory of performability of actions based on relational models and formal constructs borrowed from formal linguistics. The discussion of performability of actions is encapsulated in the form of a strict logical system  $\models$ . This system is semantically defined in terms of its intended models in which the role of actions of various types (atomic, sequential and compound ones) is accentuated. Since due to the nature of compound actions the system  $\models$  is not finitary, other semantic variants of  $\models$  are defined. The focus is on the system  $\models_f$  of performability of finite compound actions. An adequate axiom system for  $\models_f$  is defined. The strong completeness theorem is the central result. The role of the canonical model in the proof of the completeness theorem is highlighted. The relationship between performability of actions and dynamic logic is also discussed.

**Keywords** Binary relation · Frame · Model · Atomic action · Sequential action · Compound action · Performability of actions

---

This research was supported by the National Science Centre of Poland (BEETHOVEN, UMO-2014/15/G/HS1/04514).

---

✉ Janusz Czelakowski  
jczel@uni.opole.pl

<sup>1</sup> Department of Mathematics, Physics and Informatics, University of Opole, Opole, Poland

**Mathematics Subject Classification** 03B50 · 03B60 · 03B80

## 1 Introduction

The present paper is motivated by the research in the area of action theory made by Maria Nowakowska in her pioneering works (Nowakowska 1973a, b, 1979). She presented there an approach to action theory based on the apparatus of formal linguistics. Actions are structured entities. She distinguishes atomic actions and compound actions. Atomic actions are regarded as primitive entities not divisible into smaller parts in the adopted model. Symbols of atomic actions are treated as letters of an alphabet. Finite strings of atomic actions are words over the alphabet consisting of atomic actions. Each string represents a sequence (of names) of consecutively performed atomic actions usually in accordance with some instituted action plan. Compound actions are sets of words. From the formal viewpoint each compound action is a language over the alphabet of atomic actions. The above formal constructs define the syntax of actions.

She wanted her theory to be close as much as possible to everyday situations in which agents (individual or collective) performed various actions such as baking bread, cooking a dinner, sewing a dress or manufacturing a car etc. The problem is how to meaningfully interpret the expressions of the above language of actions in a consistent way. There are various options available here. The simplest one, which stems from dynamic logic, is to model atomic actions as binary relations on a set of states. Thus each atomic action  $a$  is represented as a kind of black box with a system of inputs and outputs. As the actions are often non-deterministic, the binary relations representing atomic actions need not be partial functions defined on the set of states. Consequently, one input may yield many possible outputs (results of the action). Then one has to answer other pertinent questions: what meaning should one attach to sequences  $x$  of letters representing atomic actions and to languages over the alphabet representing compound actions? The idea is to employ the operation of composition of binary relations, well known from set theory. Each sequence  $x$  of letters of atomic actions is represented by the corresponding finite sequence of binary relations. Taking the composition of the above associated sequence of relations, one defines the resultant relation of the given sequence of interpreted atomic actions. The resultant relation is also a binary relation on the set of states. It is also called the resultant action of a sequence of atomic actions; this relation represents the way the results of a string of instituted actions are obtained. This idea enables one to attach meanings to sets  $A$  of strings of atomic actions as well. This paper contains a detailed elaboration of this idea.

But this is still an oversimplified picture of action. In practice one is compelled to take into account various constraints imposed on performability of actions. These constraints are of different kind and weight: legal, physical, logical, computational etc. The situational envelope of action is yet another factor which should be taken into account. Actions are performed in a definite place and time, they involve cooperating agents; and last but not least—actions may consume various resources and cost money.

Yet another issue linked with performability is purposefulness of actions. One may argue that although such and such action is performable, its performability may not be rationally justified. E.g. one may build the tallest skyscraper in the world somewhere in Sahara, because it is logistically feasible. However no one will see any rationale for such an action. Leaving aside the issue of purposefulness of action, we shall merely discuss here a bunch of issues centered around the notion of performability of actions.

The semantic approach to action presented here is of the linguistic origin but it departs from the “standard” stit semantics. Comparisons with the stit semantics are possible if one additionally induce semi-orders on the set of states of the pertinent models and introduces agents on the stage. This issue is thoroughly discussed in Czelakowski (2015) in the context of ordered situational action systems. The logical systems presented in this work abstract from many situational and order aspects of action. The notion of performability is limited here to simple action models. In these models situations are reducible to states. Therefore the notion of performability studied in this work is simplified and it is formulated in terms of states and not of situations in general. A more general account of performability is outlined in the monograph (Czelakowski 2015).

The discussion on action performability is impersonal here (the agents of actions are omitted) and it is carried out in a simple, “bare” language of action. It should be underlined that the conception outlined in this paper is amendable to various syntactical and semantic extensions to do justice to a more complex parlance of action involving various situational components. For example, the work (Czelakowski 2019) incorporates agents to the above picture of action. The resulting “agential” action systems are examples of simple situational systems. They are built by applying context free grammars in Greibach form.

It is obvious that when one describes actions, one applies various idealization procedures—one isolates merely crucial states among those being involved in the process of performing of the undertaken action. Let us call them *nodal states*, because it is not generally feasible to grasp all states pertinent to an action. In particular, the description of atomic actions take into account such nodes as their inputs and outputs; anything else is irrelevant and omitted in the idealization procedure. In more exact approximations, atomic actions may turn into compound actions operating on more complex state spaces.

It should be added that action theory is a vibrant research area investigated from the perspective of AI, and numerous actions languages such as STRIPS, PDDL, A, B, BC, C+, MAD, etc. have been defined and studied. All such languages define known concepts in AI planning such as action, compound action, (related to) possible and realizable actions, deterministic (planning), frame models, etc. that are defined in the paper inside the framework. Basically all such action languages come with software tools (planners) that synthesize series of actions (plans), so they enjoy also the feature of having being implemented. We refer to Gelfond and Lifschitz (1998), Giunchiglia and Lifschitz (1998), Lee et al. (2013), Fikes and Nilsson (1971), Giunchiglia et al. (2004), Lifschitz and Ren (2006) for more details. The semantics of the above languages can be reconstructed within the framework of situational action systems and other concepts studied in Czelakowski (2015).

This paper presents companion results to the book (Czelakowski 2015) and to the earlier papers of the author (see e.g. Czelakowski 1996). The focus is on the issue of performability of actions from the semantic perspective. It seems that the problem of performability of actions, though of basic importance for purely practical applications, has not been elaborated in the literature in a systematic way thus far.

The paper is mainly concerned with a bunch of issues centered around the notion of performability of *compound actions* in models of action. Intuitively, each of the atomic actions forming a compound action may be individually performable in definite states, but the whole compound action may be not.

Deontological issues are not discussed here. They are presented in a separated work—see Czelakowski (2020).

The logic of performability, denoted by  $\models$ , occupies the central position. This logical system is semantically defined by means of its intended models in which the role of actions of various types (atomic, sequential and compound ones) is accentuated. This logic should not be confused with dynamic logic (DL). The relationship between DL and various variants of the logic of performability is thoroughly discussed in the paper.

Since the object language of the parlance about actions contains symbols of infinite compound actions, the consequence relation  $\models$  is not finitary. Therefore other semantic variants of  $\models$  are defined. The focus is on the system  $\models_f$  of performability of finite compound actions. An adequate axiom system for  $\models_f$  is defined. The strong completeness theorem is the central result. The role of the canonical model in the proof of the completeness theorem is underlined.

The main aspects of axiomatisation of performability are basically the following ones:

- (1) Performability of sequences of actions is closed under non-empty prefixes;
- (2) The empty set cannot be performed;
- (3) Performability of a set of sequences is the same as performability of at least one sequence belonging to this set.

## 1.1 Outline of the Paper

Section 2 briefly recalls basic linguistic notions, which reports the syntax. Section 3 is concerned with relational models of action, which basically define fluents and actions, and then the paper goes into details of actions and their performability. The paper shows in this section a number of concepts and definitions, that together lead to the goal of the paper. The focus of Sect. 4 is on logical issues. The language of action performability is defined and the logic of performability  $\models$  in the form of a semantically defined consequence relation is discussed there. Section 5 is devoted to ultrasets. The role of ultrasets and the canonical model is highlighted in Sect. 6. The subsequent section reports basic properties of the logic  $\models$ . Sections 8 and 9 deal with the logic of finite actions and the relationship of the logic of performability with dynamic logic. Section 10 discusses other issues pertinent to performability of actions as e. g. degrees of performability, fuzzy performability etc. The paper presents basic

results concerning logical aspects of performability leaving detailed applications of the presented theory for further scrutiny.

## 2 The Alphabet of Action

Let  $(\Sigma^*, \bullet, e)$  be the free semigroup freely generated by a nonempty set of generators  $\Sigma$ . Thus, the elements of  $\Sigma^*$  are finite sequences of members of  $\Sigma$ .  $\bullet$  is the operation of concatenation of sequences.  $e$  stands for the empty sequence.

The elements of  $\Sigma$  are called *symbols of atomic actions* while the elements of  $\Sigma^*$  are referred to as *sequences of symbols of atomic actions* or simply *symbols of sequential actions*. We shall simply refer to the elements of  $\Sigma^*$  as to *words*.

From the linguistic viewpoint, the set  $\Sigma$  is an alphabet and the members of  $\Sigma^*$  are (finite) words over  $\Sigma$ . (Here the set  $\Sigma$  is allowed to be infinite.)

$\wp(\Sigma^*)$  is the power set of  $\Sigma^*$ . Thus, the elements of  $\wp(\Sigma^*)$  are subsets of  $\Sigma^*$ . They are formal languages over the alphabet  $\Sigma$ . But here we adhere to the action theory terminology and call the elements of  $\wp(\Sigma^*)$  *symbols of compound actions* over  $\Sigma$ . Accordingly, from the linguistic perspective, symbols of compound actions are the same objects as languages over  $\Sigma$ .

The elements of  $\Sigma$  will be marked as  $a, b, c, d$  with indices if necessary. Sequential actions (words) will be denoted by  $x, y, z, w$  with indices if necessary. In turn, compound actions will be marked by capital letters  $A, B, C$  etc.

The algebraic structure of the semigroup  $(\Sigma^*, \bullet, e)$  is quite simple. The concatenation operation  $\bullet$  is associative (but not commutative),  $e$  is the neutral element, which means that  $x \bullet e = e \bullet x = x$ . Following common practice we shall mark the word  $x \bullet y$ , being the concatenation of the words  $x$  and  $y$ , simply as  $xy$ , suppressing the symbol  $\bullet$ . Thus, if  $x = a_1 \dots a_m$  and  $y = b_1 \dots b_n$  are words, their concatenation  $xy$  is written down as  $a_1 \dots a_m b_1 \dots b_n$ , as it is customary in formal linguistics.

$\Sigma^*$  is also endowed with the unary operation  $r$  of reflection. The operation of reflection is not discussed in this paper in the context of action performability.

**Note** Formally speaking, symbols of atomic actions are not members of  $\Sigma^*$ . But for each atomic action  $a \in \Sigma$ , the sequence  $\langle a \rangle$  of length 1 is a word. We shall identify the atomic action  $a$  with  $\langle a \rangle$  and therefore include  $\Sigma$  to the set of words  $\Sigma^*$  as a subset. Consequently, all symbols atomic actions are regarded as elements of  $\Sigma^*$ . □

The Boolean algebra  $\wp(\Sigma^*)$ , apart from being endowed with the standard set-theoretic operation of union (join)  $\cup$ , the intersection (meet)  $\cap$  and complement  $'$  together with two constants distinguished  $\emptyset$  (the empty set, the zero of  $\wp(\Sigma^*)$ ) and  $\Sigma^*$  (the unit element of the algebra), is also furnished with further operations with clear linguistic connotations. If  $A$  and  $B$  are subsets of  $\Sigma^*$  then we define:

$$A \bullet B := \{xy : x \in A, y \in B\},$$

$$A^r := \{x^r : x \in A\}.$$

The compound action  $A \bullet B$  is called the *composition of the actions A and B* and  $A^r$  is called the *reflection of A*.

Moreover, for each natural number  $n \geq 0$  we recursively define:

$$\begin{aligned} A_0 &:= \{e\} \\ A^{n+1} &:= \{xy : x \in A^n, y \in A\}, \\ A^+ &:= \bigcup_{n \geq 1} A^n, \\ A^* &:= \bigcup_{n \geq 0} A^n. \end{aligned}$$

Note that  $A^1 = A$  and  $A^* = A^+ \cup \{e\}$ .  $A^*$  is called the *Kleene closure* of  $A$ . The set  $A^+$  is the *positive closure* of  $A$ .

The so enriched Boolean algebra is denoted by  $\wp(\Sigma^*)$  and it is called the *language of compound actions*.

### 3 Relational Models of Action

**Definition 3.1** By an *action model* over  $\Sigma$  we shall understand a triple

$$M = \langle W, V, V_R \rangle$$

such that:

- (1)  $W$  is non-empty set called the *set of states*.
- (2)  $V$  is a mapping defined on  $\Sigma$  assigning to each symbol  $a \in \Sigma$  a binary relation  $V(a)$  defined on the set  $W$ .
- (3)  $V_R$  is a mapping defined on  $\Sigma$  assigning to each symbol  $a \in \Sigma$  a sub-relation  $V_R(a)$  of  $V(a)$ . □

The mapping  $V$  provides an interpretation of each symbol  $a \in \Sigma$ .  $V(a)$  is called the *action of  $a$*  on the states of  $W$ .

Any pair  $\langle u, w \rangle \in V(a)$  is called a *possible performance* of  $a$  in the model  $M$ . Accordingly, the interpretation  $V(a)$  of an action symbol  $a$  is identified with the set of possible performances of  $a$  in the model.

If  $V(a)$  is empty, the set of possible performances of  $a$  is empty.

Any pair  $\langle u, w \rangle$  belonging to  $V_R(a)$  is called a *realizable performance* of  $a$  in the model. The relation  $V_R(a)$  is therefore the set of *realizable performances* of the action  $a$ .<sup>1</sup>

As  $V_R(a)$  is a subset of  $V(a)$ , every realizable performance is possible, but generally not vice versa.

<sup>1</sup> There does not exist a uniform and coherent terminology in the literature concerning various aspects of performability of actions. In this paper possible and realizable performances of an action are distinguished. In some contexts one may also speak of *virtual* and *actual* performances of actions; in other situations, e.g., those involving games, one may contrapose performances versus successful performances etc.

The relations  $V(a)$  and  $V_R(a)$  are the key components of the notion of performability of actions in the model  $M$  we shall investigate later. Here we shall merely outline the main idea.

Let  $u \in W$  be a state. The action  $a$  is *performable at the state  $u$*  in the model  $M$  if *some* possible performance of  $a$  instituted at  $u$  is realizable, that is, there exists a state  $w$  such that  $\langle u, w \rangle$  is an actual performance of  $a$ , i.e.,  $\langle u, w \rangle \in V_R(a)$ .

The distinction between possible and realizable performances is a vital part of the theory.

Let us first give a simple example taken from everyday life. I am standing by my car in the parking lot. My intention is to open the car door and get into the car. I am the driver. I want to reach the state  $w$  in which the door is open. Possible performances of this action  $A$  lead from my current state in which the door is closed to the state  $w$ . Basically, one can isolate two initial states: the state  $u_1$  in which I have the car keys with me and  $u_2$  in which I do not have them (the keys have been lost or stolen etc.); other possibilities are omitted. Thus  $A$  consists of two possible performances:  $\langle u_1, w \rangle$  and  $\langle u_2, w \rangle$ . Suppose that  $A = V(a)$  for some action symbol  $a$ . The door gets open by the keys only in the state  $u_1$ . Accordingly,  $V_R(a)$  consists of only one pair  $\langle u_1, w \rangle$  and the action  $A$  is  $R$ -performable at  $u_1$ ; formally, because  $\langle u_1, w \rangle \in V_R(a)$ . As  $V_R(a)$  represents the “civilised” way of opening the car door by means of the keys, the action  $A$  is not performable at  $u_2$ . Thus though  $\langle u_2, w \rangle$  is a possible performance, it is not a realizable performance.

But now suppose that I am desperate and I decide to force the door, because some important documents are left in the car. The action  $A$  is the same, it consists of two possible performances  $\langle u_1, w \rangle$  and  $\langle u_2, w \rangle$ . However in this case  $V_R(a)$  is replaced by another relation, viz.,  $V'_R(a) := \{\langle u_1, w \rangle, \langle u_2, w \rangle\}$ . This is due to the fact that by applying the physical force, the door can be opened in each of the states  $u_1, u_2$ . Therefore, in this case  $A (= V'_R(a))$  is performable in both states  $u_1$  and  $u_2$ . This example explains why it is necessary to adopt various meanings of performability of the same action  $a$ , each depending on a selection of appropriate model  $M$ .

In the game of snooker, the execution of a shot by striking the cue (white) ball in a direction of a red ball is a possible performance of the action of shooting the cue ball. But only when the cue ball hits the red ball and sinks it in a pocket, this performance is successful (or realizable, according to the above parlance).

A model  $\langle W, V, V_R \rangle$  is *deterministic* if for every  $a \in \Sigma$ , the relation  $V(a)$  is a unary total function, i.e., it is a function whose domain is  $W$ . In any deterministic model, every relation  $V_R(a)$  is a partial function, being the restriction of  $V(a)$  to a non-empty subset of  $W$ .

Deterministic models play a significant role in the presented approach, because their logical strength is the same as the class of all relational models. In other words, the logical system  $\models$  we shall introduce is semantically defined by means of *all* relational models. But in view of Adequacy Theorem (Theorem 7.1),  $\models$  is characterized by a single model  $M_c$ , the canonical model of  $\models$ . The model  $M_c$  is deterministic.

**Note 3.2 Frame-based action models** Simple models of action are often defined in the following way. Suppose  $W$  is a non-empty set and  $V$  is an interpretation of  $\Sigma$  in

$W$ , that is,  $V$  is a mapping assigning to each symbol  $a \in \Sigma$  a binary relation  $V(a)$  on  $W$ .

Let  $R$  be a binary relation on  $W$ .  $R$  is called the *global transition* relation on the set of states. We then define:

$$V_R(a) := V(a) \cap R,$$

for all  $a \in \Sigma$ . Thus  $\langle W, V, V_R \rangle$  is an action model.

The structure  $\langle W, R, \{V(a) : a \in \Sigma\} \rangle$  is an atomic action system in the sense of Czelakowski (2015).

In the above example, the pair  $\langle W, R \rangle$  is called a *relational frame* (it is also known as a *Kripke frame*) and the model  $\langle W, V, V_R \rangle$  is referred to as a *frame-based action model*.

Frame-based models will be usually marked as

$$\langle W, R, V \rangle$$

to highlight the special status of the transition relation  $R$  in it. The operator  $V_R$  is then defined according to the above formulas. This assumption is tacitly assumed whenever frame-based models come to light in various contexts.

Frame-based models are indispensable in situations when one is concerned with the issue of concerted actions of agents, because the transition relation  $R$ , apart from other situational aspects, enables one to express in a uniform way principles of cooperation of agents. Constructions of models of this type are also a valuable source of examples and counter-examples to various problems pertinent to action theory—see e.g. Sect. 8.  $\square$

Definition 3.1 of an action model does not refer to a transition relation  $R$  globally defined on the set  $W$ —each relation  $V_R(a)$  is merely locally defined as a subrelation of  $V(a)$ . Therefore Definition 3.1 provides a larger class of action models than Example 3.2.

Let  $M = \langle W, V, V_R \rangle$  be an action model over  $\Sigma$ . Formally, every word  $x$  of  $\Sigma^*$  is interpreted in the model  $M$  as a finite sequence of binary relations. If  $x = a_1 \dots a_m$  is a non-empty word, then the sequence of binary relations

$$V_{seq}(x) := \langle V(a_1), \dots, V(a_m) \rangle$$

is called the *sequential action* of  $x = a_1 \dots a_m$  in the model  $M$ .

If the length of  $x$  is 1,  $x = \langle a \rangle$ , then  $V_{seq}(a) = \langle V(a) \rangle$ . Following common practice, the sequence  $\langle a \rangle$  is identified with the symbol  $a$  itself. We shall therefore identify the sequence  $V_{seq}(x)$  with the relation  $V(a)$ .

For the empty sequence  $e$ , it is assumed that  $V_{seq}(e)$  is the diagonal relation  $\mathbf{0}_W = \{\langle w, w \rangle : w \in W\}$ .

If  $A \subseteq \Sigma^*$  is a compound action, then

$$V_{seq}(A) := \{V_{seq}(x) : x \in A\}$$

is the set of all sequential actions of  $A$  in the model, or shortly, the *action of  $A$  in the model*.

**Definition 3.3** Let  $\langle W, V, V_R \rangle$  be an action model. The mapping  $V$  is inductively extended on the set  $\Sigma^*$  of words by means of the operation of composition of binary relations.

For the empty word  $e$ , the relation  $V(e)$  is the diagonal of  $W$ ,

$$V(e) = \mathbf{0}_W.$$

Hence  $V(e) = V_{seq}(e)$ .

For any word  $x \in \Sigma^*$  and any  $a \in \Sigma$ ,

$$V(xa) := V(x) \circ V(a)$$

where  $\circ$  is the composition operation of relations. Thus  $uV(xa)w$  if and only if there exists a state  $v$  such that  $uV(x)v$  and  $vV(a)w$ .

Accordingly, if  $x = a_1 \dots a_m$  is a non-empty word, then

$$V(x) = V(a_1) \circ \dots \circ V(a_m),$$

i.e.,  $uV(x)w$  if and only if there exists a sequence of states  $u_1 \dots u_m$  with  $u_m = w$  such that

$$uV(a_1)u_1V(a_2)u_2 \dots u_{m-1}V(a_m)u_m.$$

The binary relation  $V(a_1 \dots a_m)$  is called the *resultant action of the sequence*  $V_{seq}(a_1 \dots a_m) = \langle V(a_1), \dots, V(a_m) \rangle$  on the states of  $W$ .

In the third step,  $V$  is extended onto any subset  $A \subseteq \Sigma^*$ :

$$V(A) := \bigcup \{V(x) : x \in A\}.$$

The binary relation  $V(A)$  is called the *resultant action of the set*  $V_{seq}(A)$  of sequential actions of  $A$  in the model. □

$V(A)$  is a binary relation on  $W$ .  $V(\emptyset)$  is the empty set. If  $A = \Sigma^*$ , then  $V(\Sigma^*)$ , the set-theoretic union of the relations  $V(x)$ ,  $x \in \Sigma^*$ , may be a proper binary relation on  $W$ .

It is clear that  $V(x) \circ V(e) = V(e) \circ V(x) = V(x)$ , for all words  $x$ .

To define the notion of *performability* of sequential and compound actions, we shall suitably modify the above definition. As it was said above, if  $a \in \Sigma$ , the pairs of states belonging to  $V_R(a)$  are called *realizable performances* of  $V(a)$  in the model.  $V_R(a)$  is called the *realizable resultant action* of  $V(a)$  in the model.

**Definition 3.4** In an analogous manner the mapping  $V_R$  is extended onto the words of  $\Sigma^*$ . According to the definition of  $V_R$ , the relation

$$V_R(e)$$

is as an *arbitrary* but fixed subset of the diagonal relation  $V(e)$ .

Then, for any *non-empty* word  $x \in \Sigma^*$  and any  $a \in \Sigma$ , we put:

$$V_R(xa) := V_R(x) \circ V_R(a).$$

Thus  $uV_R(xa)w$  if and only if there exists a state  $v$  such that  $uV_R(x)v$  and  $vV_R(a)w$ .

It follows that if  $x = a_1 \dots a_m$  is a non-empty word, then  $uV_R(x)w$  holds if and only if there is a sequence of states  $u_1 \dots u_m$  with  $u_m = w$  such that

$$uV_R(a_1)u_1V_R(a_2)u_2 \dots u_{m-1}V_R(a_m)u_m.$$

**Note** As  $ex = x = xe$  for any word  $x$ , we obviously have that  $V_R(ex) = V_R(xe) = V_R(x)$ . But it need *not* be the case that  $V_R(e) \circ V_R(x) = V_R(ex)$  and  $V_R(x) \circ V_R(e) = V_R(xe)$ .  $\square$

The binary relation  $V_R(a_1 \dots a_m)$  is therefore defined for all words  $x = a_1 \dots a_m \in \Sigma^*$ .  $V_R(x)$  is called the *realizable resultant relation of the sequence  $V_{seq}(x)$  in the model*.

If  $uV_R(x)w$  holds for  $x = a_1 \dots a_m$ , we say that the pair  $\langle u, w \rangle$  is a *resultant realizable performance* of the sequence of actions  $V_{seq}(x) = \langle V(a_1), V(a_2), \dots, V(a_m) \rangle$ .

For any set  $A \subseteq \Sigma^*$  we define:

$$V_R(A) := \bigcup \{V_R(x) : x \in A\}.$$

The binary relation  $V_R(A)$  is called *the realizable resultant action of the compound action  $A$* . Trivially,

$$V_R(\emptyset) = \emptyset.$$

If  $uV_R(A)w$  holds, the pair  $\langle u, w \rangle$  is called a *resultant realizable performance of  $V(A)$* .  $\square$

## Performability

The paper offers a conception of performability of actions formulated in terms of the above action models. It slightly differs from the one presented in the monograph (Czelakowski 2015). There one can also find a justification of this conception. But the focus of this work is on logical aspects of performability. This issue is not thoroughly discussed in the above monograph.

We adopt the following definitions:

**Definition 3.5** Let  $M = \langle W, V, V_R \rangle$  be a model and  $u \in W$  a state.

- (1) Let  $a$  be in  $\Sigma$ . The atomic action  $V(a)$  is *performable at  $u$*  if and only if there exists a state  $w$  such that  $uV_R(a)w$  holds.

- (2) Let  $x$  be in  $\Sigma^*$ . The sequential action  $V_{seq}(x)$  is *performable at  $u$*  if and only if there exists a state  $w$  such that  $uV_R(x)w$  holds.  
 In particular, the action  $V_{seq}(e) (= V(e))$  is performable at  $u$  if and only if it is the case that  $uV_R(e)u$ .
- (3) Let  $A$  be a subset of  $\Sigma^*$ . The compound action  $V_{seq}(A)$  is *performable at  $u$*  if and only if there exists a state  $w$  such that  $uV_R(A)w$  holds.  
 Thus  $V_{seq}(A)$  is performable at  $u$  if and only if for *some* word  $x \in A$ , the sequential action  $V_{seq}(x)$  is performable at  $u$ , i.e.,  $uV_R(x)w$  for some state  $w$ . □

The above definition provides a *weak* notion of performability of actions. We shall denote it for brevity as  $\exists$ -performability. This notation is motivated by the use of the existential quantifier “there exists” in the *definienses* of definitions (1)–(3). Performability of actions is thus a version of the modal possibility operator (cf. Sect. 9.)

Some comments to the above definitions. According to (1),  $V(a)$  is performable at  $u$  in the sense of  $M$  if and only if for some state  $w$ , the pair  $\langle u, w \rangle$  is a realizable performance of  $V(a)$ , i.e,  $\langle u, w \rangle \in V_R(a)$ . Thus  $\langle u, w \rangle$  is a possible performance of  $A$  and the transition from  $u$  to  $w$  is admissible by  $V_R(a)$ . Analogously, according to (2), if  $x = a_1 \dots a_m$ ,  $V_{seq}(x)$  is performable at  $u$  if and only if for some state  $w$ , the pair  $\langle u, w \rangle$  is a resultant realizable performance of  $V_{seq}(x)$ , which means that there is a sequence of states  $u_1 \dots u_m$  with  $u_m = w$  such that  $uV_R(a_1)u_1 \dots u_{m-1}V_R(a_m)u_m$ .

It follows from the above definitions that if  $V_{seq}(xa)$  is performable at  $u$ , then  $V_{seq}(x)$  is performable at  $u$  as well, for all non-empty words  $x$  and symbols  $a$ . Consequently, if  $V_{seq}(x)$  is performable at  $u$  then so are all non-empty prefixes of  $x$ .

The problem of performability of compound actions is a difficult part of action theory. How to explicate it in simple terms and present a solution in a conceptually incontrovertible way? Definition (3) offers an approach to this problem (see also Note below). According to (3),  $V_{seq}(A)$  is performable at  $u$  if merely *some* word  $x$  in  $A$  is performable at  $u$ , i.e., for some word  $x \in A$  there is a state  $w$  such that  $\langle u, w \rangle$  is a resultant realizable action of  $V_{seq}(x)$ .

Consequently, the action  $V_{seq}(A)$  is unperformable at a state  $u$  if and only if *all* sequential actions  $V_{seq}(x)$ ,  $x \in A$ , are unperformable at  $u$ .

The compound action  $\emptyset$  is never performable.

**Definition 3.6** Let  $M = \langle W, V, V_R \rangle$  be a model over  $\Sigma$ .

For each  $a \in \Sigma$  we define the *performability proposition Perf*  $V_{seq}(a) \subseteq W$ :

$$(26.a) \quad u \in \mathbf{Perf} V_{seq}(a) \Leftrightarrow_{df} V_{seq}(a) \text{ is performable at } u \\ \Leftrightarrow (\exists w \in W) uV_R(a)w.$$

This definition is extended onto arbitrary words  $x \in \Sigma^*$ :

$$(26.b) \quad u \in \mathbf{Perf} V_{seq}(x) \Leftrightarrow_{df} V_{seq}(x) \text{ is performable at } u \\ \Leftrightarrow (\exists w \in W) uV_R(x)w.$$

In particular,

$$\begin{aligned} u \in \mathbf{Perf} V_{seq}(e) &\Leftrightarrow_{df} V(e) \text{ is performable at } u \\ &\Leftrightarrow uV_R(e)u. \end{aligned}$$

It follows from the above definitions that for any *non-empty* word  $x \in \Sigma^*$  and any symbol  $a \in \Sigma$ ,  $u \in \mathbf{Perf} V_{seq}(xa)$  if and only if there exist states  $v$  and  $w$  such that  $uV_R(x)v$  and  $vV_R(a)w$ .

Finally, for every set  $A \subseteq \Sigma^*$  we put:

$$\begin{aligned} (26.c) \quad u \in \mathbf{Perf} V_{seq}(A) &\Leftrightarrow_{df} V_{seq}(A) \text{ is performable at } u \\ &\Leftrightarrow (\exists x \in A) V_{seq}(x) \text{ is performable at } u \\ &\Leftrightarrow (\exists x \in A) u \in \mathbf{Perf} V_{seq}(x). \quad \square \end{aligned}$$

From the viewpoint of the theory of relations, the proposition  $\mathbf{Perf} V_{seq}(a)$  coincides with the domain of the relation  $V_R(a)$  for every symbol  $a \in \Sigma$ , because the domain of  $V_R(a)$  is equal, by definition, to the set  $\{u : (\exists w \in W)uV_R(a)w\}$ . Likewise,  $\mathbf{Perf} V_{seq}(x)$  is the domain of the binary relation  $V_R(x)$ , for all words  $x \in \Sigma^*$ , and  $\mathbf{Perf} V_{seq}(A)$  is the domain of the relation  $V_R(A)$ , for all sets  $A \subseteq \Sigma^*$ .

**2.7 Comments on the notion of performability A.** We shall refer to the notion of performability of compound actions  $A$  in the sense of the above definition as  $\exists$ -*performability*, due to the occurrence of the existential quantifier  $\exists$  in the *definiens*. We may mark it as  $\mathbf{Perf} \exists$ .

We point out other options according to which one may define the operator of performability of compound actions. E.g., one may argue that a compound action  $V_{seq}(A)$ ,  $A \subseteq \Sigma^*$ , is *performable in a state*  $u$  if *all* sequential actions  $V_{seq}(x)$ ,  $x \in A$ , are performable in  $u$ . This form of performability is referred to as the  $\forall$ -*performability* due to the occurrence of the universal quantifier  $\forall$  in the *definiens*. If performability is viewed as the  $\forall$ -*performability*, formula (3.6.c) turns into

$$\mathbf{Perf} \forall V_{seq}(A) = \bigcap \{\mathbf{Perf} V_{seq}(x) : x \in A\}.$$

We have therefore two notions of performability of compound actions available. Which option is right?

We consider the compound action  $A$  termed *Doing shopping in a mall* by a definite person. The selection of the mall is crucial. (He/she lives in a big city.) This action consists of finite sequences of simpler actions. This compound action is performed in initial states which are conventionally named “entering a mall” and ends up in a state in which the shopping is finished. Various scenarios of doing shopping are available. Usually one selects a number of stores and services to visit. But these selected stores may be visited in various orders. e.g. first going to the bookstore, then to the drugstore and subsequently to the liquor store. In other words, instead of *one* selected sequential action  $x$  in  $A$  representing a list of consecutive stores he/she plans to visit, it is possible to make rearrangements in the list resulting in other operational sequences, being some

permutations of the word  $x$ . This permutations of  $x$  may be also performable in a given state at which  $x$  performable.

Here performability of shopping falls under formula (3.6.c) – it is  $\exists$ -performability, because to perform this compound action is suffices to perform one string of atomic actions belonging to it and, more importantly, not all conceivable scenarios of doing shopping, i.e., not all words this compound action encompasses are realizable at a given state.

It would be an absurdity to claim that the agent performs *all* physically feasible or meaningful sequences of simple actions the shopping is composed of. It may happen that some strings of actions may be unperformable for him. E.g. even if he decides to do shopping consecutively in two or three deliberately chosen shopping centers, some other malls may be excluded, because they are too expensive or too distant.

On the other hand, let  $a$  be a fixed action symbol. Let  $A$  be the Kleene closure  $\{a\}^*$  of  $\{a\}$ , i.e.,  $A = \{a\}^* = \{a^n : n \geq 0\}$ .  $A$  is a compound action. If  $M = \langle W, V, V_R \rangle$  is a model, one may ask when  $V_{seq}(A)$  is performable.  $V_{seq}(A) = \bigcup \{V_{seq}(a^n) : n \geq 0\}$  is the set of all iterations of the atomic action  $V(a)$ . Intuitively, in this case one may argue that the performability of  $V_{seq}(A)$  is tantamount to the performability of *all* consecutive iterations  $V_{seq}(a^n)$ . Thus it is  $\forall$ -performability.

We are thus facing a dilemma—which of the two options pertinent to performability of compound actions to choose—the weak one, the  $\exists$ -performability, or the strong one, the  $\forall$ -performability?

If one computes consecutive digits in the decimal expansion of number  $\pi$ , one performs iterations of a certain atomic action represented by the symbol  $a$ . The states are represented by the successive decimal approximations to  $\pi$ . Thus  $w_0 = 3, w_1 = 3.1, w_2 = 3.14$  etc.  $W := \{w_n : n \geq 0\}$  is the set of states. The interpretation  $V(a)$  of  $a$  is the total function throughout  $W$  given by  $V(a)(w_n) = w_{n+1}$ , for all  $n \geq 0$ . Then  $V(a^n)(w_0) = w_n$  for all  $n$ .  $V_R(a)$  is declared to be equal to  $V(a)$ . Consequently,  $V(a) (= V_{seq}(a))$  is performable in each state  $w_n$ : if  $w_n$  is computed, then  $w_{n+1}$  is also computable. (If one assumes that  $V_R(a)$  terminates at some state  $w_n$ , i.e.,  $w_n$  is a dead state of the relation  $V_R(a)$ , then the computation  $V(a)$  halts in  $w_n$ .) Let  $A := \{a^n : n \geq 0\}$ . Then  $V(A)$  is the Kleene closure of the relation  $V(a)$ , i.e.,  $V(A) = \bigcup \{V(a^n) : n \geq 0\}$ . Hence  $w_0 V(A) w$  if and only if  $w = w_n$  for some  $n \geq 0$ . Computing the exact value of  $\pi$  would require performing all iterations  $a^n$  at  $w_0$  resulting in an infinite sequence  $3.14 \dots$ , an irrational number, which is *not* a state of  $W$ . Definition 3.5.(3) may seem to do not adequately capture this situation—in view of this definition to perform the compound action  $A = \{a^n : n \geq 0\}$  at  $w_0$  it merely suffices to perform one iteration  $a^n$ . But  $V_{seq}(A)$  is *not* the action of computing  $\pi$ —this is the action  $V(a)$ ! Note that  $V_{seq}(A)$  is performable at  $w_0$  in the strong sense— $A$  is  $\forall$ -performable here, because all iterations  $a^n$  are performable at  $w_0$ .  $V_{seq}(A)$  is the action of performing any randomly selected iteration of  $a$  at  $w_0$ . Although Definition 3.5.(3) does not exclude the situations in which a compound action is actually  $\forall$ -performable, we shall take care of situations when one is interested in performing iterations of actions. The presented here definitions of performability of compound actions—the  $\exists$ -performability and  $\forall$ -performability, though logically correct, being the result of straightforward applications of the general or existential quantifier in the *definiens*, do not deeply penetrate the structure of compound actions. Definition 3.5.(3)

treats compound actions as unstructured sets of words, thus it abstracts from possible interrelations holding between these words.

Yet another option which appears here takes into account the *algebraic* structure of compound actions in the *definiens* of the definitions of performability. This option will be discussed in Sect. 9, where it will be shown how to accomplish this task in case of regular actions, i.e., the actions corresponding to regular expressions. But in this and the subsequent section we shall adhere to Definition 3.5.(3) and present a uniform theory of performability based on it. We shall also discuss some other consequences this definition bears.

Due to computational limitations it is not possible to compute all successive digits in the expansion of  $\pi$ . Thus not all iterations of  $V_{seq}(a^n)$  are feasible at  $w_0$ ; one stops at some step  $n$  and computes only iterations resulting in the rational approximation of  $\pi$  up to  $10^{-n}$ . Although the action of determining *all* digits in the expansion of  $\pi$  may be regarded as  $\forall$ -performable from the logical viewpoint, it is merely  $\exists$ -performable (up to sufficiently large  $n$ -th iteration) if one takes the above limitations. We may accordingly modify the definition of  $V_R(a)$  requiring e.g. that  $V_R(a)$  is a proper subrelation of  $V(a)$  such that  $w_n$  is a dead state of  $V_R(a)$  for a sufficiently large  $n$ . Thus there is no way of leaving  $w_n$ —the computation terminates at  $w_n$ .

This and similar issues concerning unbounded iterations give rise to questions about the role of infinity in performing compound actions. There are several ways of tackling this problem. One may introduce ordered action systems in which the sets of states are ordered, satisfying a form of order completeness, and actions are order-continuous. One may then apply the methods of fixed-point theory. The results of unbounded iterations of an action  $V(a)$  result in reflexive points of the relation  $V_R(a)$ . The other option is to apply the methods of algebraic posets and domain theory, and approximate compound actions by their compact subactions. Some of these problems are discussed in Czelakowski (2015).

The two definitions of performability of compound actions—the  $\exists$ -performability, adopted in the paper, and the strong one—the  $\forall$ -performability, are based on a straightforward applications of the general or existential quantifier. They constitute, from the logical viewpoint, two extreme solutions of the problem of performability. One may argue that these solutions are incapable to express in a satisfactory way the issue of approximating a compound action  $A$  by finite or, more generally, compact subactions of  $A$ . This issue is vital when one is concerned with the necessity of halting a potentially infinite computing after a finite number of steps. In the simplest case, when  $a$  is an atomic action symbol, and  $A$  is the positive Kleene closure of  $\{a\}$ ,  $A = \{a^n : n \geq 1\}$ , then the compound action  $A$  is  $\exists$ -performable in a model at some state  $u$  if and only if merely *some* word  $a^n$ ,  $n \geq 1$ , is performable in the model at  $u$ , which may seem to be a too weak form of performability of  $A$  when one is concerned with a more accurate computation linked with  $A$ .

In turn, the  $\forall$ -performability of  $A$  at  $u$  is tantamount to performability of *all* words  $a^n$  at  $u$ , which seems to be a too strong form of performability of  $A$  in the above context.

Suppose however that  $a$  is interpreted in a model  $M$  as a serial relation  $V(a)$  on the set of states  $W$ . Moreover it is assumed that  $V(a) = V_R(a)$ . Then all words  $x = a^n$ ,  $n \geq 1$ , are performable at any state  $u \in W$ . It then follows that for any state  $u \in W$ ,

the compound action  $A = \{a^n : n \geq 1\}$  is  $\exists$ -performable at  $u$  in  $M$  if and only if there is  $m \geq 1$  such that *all* sequential actions  $a^n, n \geq m$ , are performable at  $u$  in  $M$ . This equivalence shows that the compound action  $A$  is approximated at any state  $u$  in the model by all sequential actions  $a^n$  of a sufficiently large length  $n$ . This shows that the notion of  $\exists$ -performability of compound actions is applicable in the problem of approximating various algorithms corresponding to  $a$  by finite sequences of computations with an arbitrary accuracy.

We may say more—in some models,  $\exists$ -performability coincides with  $\forall$ -performability. The following example illustrates this situation and sheds more light on the problem of performability.

Let  $\mathbf{r} = (r_n)$  be an arbitrary infinite sequence of rational numbers whose exact values are unknown. (We may assume that the sequence  $\mathbf{r}$  is convergent to a real number.) Let  $W_0$  be the set of all finite non-empty initial segments of  $\mathbf{r}$ , i.e.,

$$W_0 := \{(r_1, \dots, r_n) : n = 1, 2, \dots\}.$$

For  $u, w \in W_0$  we put:

$$u \leq_0 w \Leftrightarrow_{df} u \text{ is a prefix of } w.$$

$(W_0, \leq_0)$  is a linear poset with  $\mathbf{0} := \langle r_1 \rangle$  being the least element.

By adjoining to  $W_0$  the sequence  $\mathbf{r}$  as the supremum of the poset  $(W_0, \leq_0)$ , one obtains an algebraic poset with  $W_0$  being the set of compact elements and  $\mathbf{r}$  as the top element. The elements of  $W_0$  are also called *compact approximations* of  $\mathbf{r}$ .

Suppose that an algorithm  $a$  is available enabling to recursively compute the number  $r_1$  and, for every  $n \geq 1$ , given the sequence of values  $\langle r_1, \dots, r_n \rangle$ , establish the value of  $r_{n+1}$ .

$V(a)$  is the atomic action on the set  $W_0$  of states being the interpretation of  $a$ . Let  $w_n := \langle r_1, \dots, r_n \rangle$  for  $n = 1, 2, \dots$ .  $V(a)$  is the total unary function defined on  $W_0$  by the formula:

$$V(a)(w_n) := w_{n+1}, \text{ for } n = 1, 2, \dots$$

We also put  $V_R(a) := V(a)$ . The triple  $M = \langle W_0, V(a), V_R(a) \rangle$  is a deterministic action model over the alphabet  $\Sigma = \{a\}$  consisting of the symbol  $a$ —the name of this algorithm. The atomic action  $V(a)$  corresponding to  $a$  is performable at every state  $w_n$ , because  $V_R(a) = V(a)$ . (In practice, due to computational limitations, the function  $V_R(a)$  is “trimmed” for large values of  $n$  and replaced by the partial function  $V_R^*(a)$  which agrees with  $V_R(a)$  on the initial states  $w_1, \dots, w_n$  and is undefined for larger states  $w_{n+1}$  etc. This implies that in such a “trimmed” model the action  $V(a)$  is performable merely in the states  $w_1, \dots, w_n$  and it halts at  $w_{n+1}$ .)

Let  $A := \{a^n : n \geq 1\}$  be the positive Kleene closure of the language  $\{a\}$ . Each word  $a^n$  is interpreted in the model as the sequence  $V_{seq}(a^n) := \langle V(a), \dots, V(a) \rangle$  with  $V(a)$  repeated  $n$  times. The compound action  $A$  interpreted in the model  $M$  as

the set of sequential actions  $V_{seq}(a^n)$ :

$$V_{seq}(A) := \{V_{seq}(a^n) : n \geq 1\}.$$

It follows from the definition of  $M$ , that every sequential action  $V_{seq}(a^n)$  is performable at any state  $u \in W_0$ , because  $V_R(a^n)(w_k) = w_{k+n}$ , for all  $k, n \geq 1$ . Consequently, the compound action  $V_{seq}(A)$  is  $\forall$ -performable at any state  $u \in W_0$ , because  $\forall$ -performability of  $V_{seq}(A)$  at  $u$  means that all sequential actions  $V_{seq}(a^n)$ ,  $n \geq 1$ , are performable at  $u$ . Hence  $V_{seq}(A)$  is  $\exists$ -performable in any state  $u \in W_0$ . Thus the definitions of  $\forall$ -performability of  $A$  and  $\exists$ -performability of  $A$  coincide in this model.

As  $V_R(a^n)(\mathbf{0}) = w_{n+1}$ , for all  $n$ , the sequence  $\mathbf{r}$  is approximated in  $M$  by way of iterating the action  $V(a)$  an arbitrary number of times (commencing with the state  $\mathbf{0}$ ). Hence the limit of  $\mathbf{r}$  is approximated with an arbitrary degree of accuracy. As mentioned, in practice one must stop calculation after some number of steps by passing to the above “trimmed” model endowed with the partial function  $V_R^*(a)$ .

In a more abstract setting, the above situation is typical for models  $M = \langle W, V, V_R \rangle$  in which  $V(a)$  is a serial relation for all all words  $a \in \Sigma$  and  $V(a)$  and  $V_R(a)$  coincide on all words  $a \in \Sigma$  and on the empty word  $e$ . This is due to the fact that every sequential action  $x$  is performable in  $M$  in all possible states of  $W$ . It follows that the  $\forall$ -performability coincides with the  $\exists$ -performability for any non-empty compound action  $A$  in each such a model.

The models of the above type are first-order definable and the consequence operation determined by the class of such models validates the formulas  $Perf(a)$  and, more generally, it validates all fomulas  $Perf(x)$  for any sequential action  $x$  (cf. Sect. 7.1).

Yet another option is to assume that in realistic models of action the relations  $V_R(a)$  are all inversely well-founded, which means that there do *not* exist infinite sequences of states  $w_0, w_1, \dots, w_n, w_{n+1}, \dots$  such that  $w_n V_R(a) w_{n+1}$  for all  $n$ . This option excludes reflexive points of  $V_R(a)$  and, more generally, finite cycles of states.

The presence of unbounded iterations of actions exercises an impact on the logic of performability we shall define in the next section, resulting in the non-finitary character of this logic. The logic of performability of actions will be semantically defined in terms of action systems.

**B.** Apart from the propositions  $Perf V_{seq}(x)$  and  $Perf V_{seq}(A)$ ,  $x \in \Sigma^*$ ,  $A \subseteq \Sigma^*$ , one may also define their “reversed” counterparts, marked as  $Perf^{\leftarrow} V_{seq}(x)$  and  $Perf^{\leftarrow} V_{seq}(A)$ , respectively. We put:

$$\begin{aligned} w \in Perf^{\leftarrow} V_{seq}(x) &\Leftrightarrow_{df} (\exists u \in W) u V_R(x) w, \\ w \in Perf^{\leftarrow} V_{seq}(A) &\Leftrightarrow_{df} (\exists x \in A) w \in Perf^{\leftarrow} V_{seq}(x). \end{aligned}$$

$w \in Perf^{\leftarrow} V_{seq}(x)$  is read “ $x$  has been performed in some state so that the state  $w$  is achieved”.

Analogously,  $w \in Perf^{\leftarrow} V_{seq}(A)$  is read “ $A$  has been performed in some state so that the state  $w$  is achieved”. □

**Example 2.8**  $\mathbb{R}^n$  is the linear space consisting of all  $n$ -tuples of real numbers. The vectors  $u$  of  $\mathbb{R}^n$  are states.  $L(\mathbb{R}^n)$  is the algebra of all square  $n \times n$ -matrices over  $\mathbb{R}^n$ . Each matrix  $A = [a_{ij}]$  is a unary operation from  $\mathbb{R}^n$  to  $\mathbb{R}^n$  and therefore it is a deterministic action on the set of states  $\mathbb{R}^n$ . Let  $R = [r_{ij}]$  be a distinguished square matrix called the *transition matrix* on  $\mathbb{R}^n$ . Let  $V$  be a mapping assigning to each action symbol  $a \in \Sigma$  a matrix  $V(a) \in L(\mathbb{R}^n)$ .  $V(e)$  is the identity operation. The triple

$$\langle \mathbb{R}^n, R, V \rangle$$

is a frame-based action model.

We define, for each  $a \in \Sigma$ , the subspace  $X(a)$  of  $\mathbb{R}^n$ :

$$X(a) := \{u \in \mathbb{R}^n : V(a)(u) = R(u)\}.$$

$V_R(a)$  is, by definition, the set of all pairs of vectors  $\langle u, w \rangle$  such  $w = V(a)(u) = R(u)$ . Thus  $V_R(a)$  is the matrix being the restriction of  $V(a)$  to  $X(a)$ , i.e.,  $V_R(a) : X(a) \rightarrow \mathbb{R}^n$  and

$$V_R(a)(u) = V(a)(u), \quad \text{for all } u \in X(a).$$

It is assumed that  $V_R(e)$  is the restriction of the identity matrix to the subspace  $X(e) := \{u \in \mathbb{R}^n : u = R(u)\}$ .

$M = \langle \mathbb{R}^n, V, V_R \rangle$  is a model in the sense of Definition 3.1.  $M$  is deterministic, because all its actions  $V(a)$  are total functions defined throughout the set  $W$ . ( $R$  is also a total function.) Each  $V_R(a)$  is a partial function, the restriction of  $V(a)$  to the subspace  $X(a)$ .

The mapping  $V$  is extended onto arbitrary words  $x \in \Sigma^*$  in accordance with Definition 3.3 by applying the operation of multiplication of matrices. It follows that for any words  $x$  and  $y$ ,  $V(xy) = V(x) \circ V(y)$ , where  $\circ$  is the operation of multiplication of matrices. (The convention that  $(F \circ G)(x) = G(F(x))$  for the composition  $F \circ G$  of functions is applied here.) In a similar manner, applying Definition 3.4, one extends  $V_R$  onto arbitrary words  $x$ .

The action  $V(a)$  is performable (in the sense of  $M$ ) exactly in the states  $u \in X(a)$ . A sequential action  $V_{seq}(a_1 \dots a_m) = \langle V(a_1), \dots, V(a_m) \rangle$ ,  $m \geq 1$ , is performable at  $u$  if and only if  $V(a_1 \dots a_m)(u) = V_R(a_1 \dots a_m)(u)$ . The action  $V(e)$  is performable at  $u$  if and only if  $u = R(u)$ , i.e., the vector  $u$  is a fixed point of  $R$ .

Suppose that the matrix  $R$  is symmetric (i.e.,  $R$  is equal to its transpose) and convertible.  $R$  is then represented in the diagonal form so that the eigenvectors  $u_i$ ,  $i = 1, \dots, n$ , of  $R$  form a basis of  $\mathbb{R}^n$  with  $\lambda_i$  being the corresponding real eigenvalue, i.e.,  $R(u_i) = \lambda_i u_i$ , for  $i = 1, \dots, n$ . Then  $V(a)$  is performable at  $u_i$  if and only if  $V(a)(u_i) = \lambda_i u_i$ , for  $i = 1, \dots, n$ . □

**Example 2.9** More general, as compared with Example 2.8, deterministic models of action are defined in terms of linear and bounded operations on normed spaces.

Let  $\langle X, \|\cdot\| \rangle$  be a normed linear space. The vectors of  $X$  are treated as states. Thus  $X$  is the set of states. Let  $V$  be a mapping assigning to each action symbol  $a \in \Sigma$

a bounded linear operation  $V(a) \in L(X)$ .  $V(e)$  is the identity operation. Let  $R$  be a distinguished bounded linear operation called the *transition operation* on  $X$ . The triple

$$\langle X, R, V \rangle$$

is a frame-based, deterministic action model.

We define, for each  $a \in \Sigma$ , the closed subspace  $X(a)$  of  $X$ :

$$X(a) := \{u \in X : V(a)(u) = R(u)\}.$$

$V_R(a)$  is the linear operation being the restriction of  $V(a)$  to  $X(a)$ :

$$V_R(a)(u) := V(a)(u) (= R(u)), \text{ for all } u \in X(a).$$

$V_R(a)$  maps  $X(a)$  to  $X$ .

The triple

$$M = \langle X, V, V_R \rangle$$

is an action model in the sense of Definition 3.1.

The mapping  $V$  is extended onto arbitrary words  $x \in \Sigma^*$  in accordance with Definition 3.3 by applying the operation of composition of linear operations and assuming that  $V(e)$  is the identity operation. Thus, for any words  $x$  and  $y$ ,  $V(xy) = V(x) \circ V(y)$ , where  $\circ$  marks the composition.

In a similar manner, applying Definition 3.4, one extends  $V_R$  onto arbitrary words  $x$ . It is assumed that  $V_R(e)$  is the restriction of the identity matrix to the subspace  $X(e) := \{u \in X : u = R(u)\}$ . Note that the domain of the linear operator  $V_R(ab)$  is equal to  $\{u \in X : V(a)(u) = R(u) \text{ and } V(b)(V(a)(u)) = R(V(a)(u))\} (= \{u \in X : V(a)(u) = R(u) \text{ and } V(b)(R(u)) = R(R(u))\})$ . This formula extends onto arbitrary words  $x = a_1 \dots a_m$  in a straightforward way.

The action  $V(a)$  is performable (in the sense of  $M$ ) exactly in the states  $u \in X(a)$ . A nonempty sequential action  $V_{seq}(a_1 \dots a_m) = \langle V(a_1), \dots, V(a_m) \rangle$ ,  $m \geq 1$ , is performable at  $u$  if and only if  $V(a_1 \dots a_m)(u) = V_R(a_1 \dots a_m)(u)$ . The action  $V(e)$  is performable at  $u$  if and only if  $u = R(u)$ .  $\square$

## 4 The Language of Performability of Actions and its Semantics

We first define the language  $L$  of action performability. We touch here a delicate syntactical issue. Firstly, the syntax of the language we shall define is as close as possible to the language of formal linguistics that is employed in automata theory. The syntax of  $L$  departs from the standard grammatical rules of forming terms and formulas adopted in contemporary logic. It is a propositional language (and thus quantifier free) but it based on atomic propositional formulas of two sorts.

Atomic formulas are expressions of the form:

$$\begin{aligned} & Perf(x), \quad x \in \Sigma^*, \\ & Perf(A), \quad A \text{ is a subset of } \Sigma^*. \end{aligned}$$

As each symbol  $a \in \Sigma$  is identified with a word of length 1, all formulas of the form  $Perf(a)$ ,  $a \in \Sigma$ , are included in the set of formulas  $\{Perf(x) : x \in \Sigma^*\}$ . The formula  $Perf(e)$  is also atomic.

$\Sigma$  is a finite or countably infinite set of variables representing atomic actions. ( $\Sigma$  is the alphabet of actions.)  $\Sigma^*$  is the set of all finite sequences of atomic actions variables. The elements of  $\Sigma^*$  are called *sequences* or *words* of atomic action variables. As the set  $\Sigma^*$  is countably infinite, there are uncountably many formulas of the form  $Perf(A)$  with  $A \subseteq \Sigma^*$ .

Compound formulas are built from the above atomic formulas (of any sort) by means of applying the Boolean connectives  $\rightarrow$  and  $\neg$ . The other Boolean connectives are defined in the standard way.

Note that there are no propositional variables. Thus the above language defines Boolean interrelations holding between action performability formulas only.  $L$  is incapable of expressing conditionals like: *If it the case that  $\Phi$ , then the action  $A$  is performable*.  $L$  does not take into account situational components of action like time, space locations, agents and their interrelations etc. Though  $L$  is simple, it makes it possible to conduct a discourse on actions that involve phrases of type: *If the action  $A$  is performable, then  $B$  is unperformable* etc.

### 4.1 The Semantics for $L$

The meanings attached to the formulas of  $L$  are defined in terms of the above action models. We define the notion of truth of formulas.

Let  $M = \langle W, V, V_R \rangle$  be a model and  $u$  a state in  $W$ . The notation

$$M, u \models \sigma$$

means that  $\sigma$  is true in the model  $M$  at  $u$ .

**Definition 4.1** Let  $x$  be in  $\Sigma^*$ . Then

$$M, u \models Perf(x) \Leftrightarrow_{df} u \in \mathbf{Perf} V_{seq}(x),$$

i.e., the sequential action  $V_{seq}(x)$  is performable at  $u$  in  $M$ . (In particular, if  $a \in \Sigma$ , then

$$M, u \models Perf(a) \Leftrightarrow_{df} u \in \mathbf{Perf} V(a).)$$

Let  $A$  be a subset of  $\Sigma^*$ . Then

$$M, u \models Perf(A) \Leftrightarrow_{df} u \in \mathbf{Perf} V_{seq}(A),$$

i.e., the compound action  $V_{seq}(A)$  is performable at  $u$  in  $M$ . □

The definition of  $M, u \models$  is extended onto compound formulas as in classical logic. Thus

$$M, u \models \phi \rightarrow \psi \text{ if and only if it is not the case that } M, u \models \phi \text{ or } M, u \models \psi;$$

and

$$M, u \models \neg\phi \text{ if and only if it is not the case that } M, u \models \phi.$$

It should be noted that the language  $L$  is too poor so that it could recursively express the definition of  $Perf(x)$  in terms of the constituents  $Perf(a)$ , where  $a$  occurs in  $x$ . E.g. it is not possible to define  $Perf(xa)$  in terms of  $Perf(x)$  and  $Perf(a)$  without resorting to the external operation of composition of relations. The symbol representing composition is absent in the vocabulary of  $L$ .

$Perf$  may be treated as the alethic operator of possibility due to meaning attached to the formulas  $Perf(x)$  in Definitions 3.5–3.6, for all  $x \in \Sigma^*$ . The structure of the language  $L$  departs, however, from the standard constructions adopted in multi-modal or dynamic logics.

A formula  $\sigma$  is *true in* in the model  $M = \langle W, V, V_R \rangle$ , in symbols:

$$M \models \sigma,$$

if and only if  $M, u \models \sigma$  for all  $u \in W$ ; we then also say that  $\sigma$  is *valid in*  $M$ .

$\sigma$  is *logically valid* if it is valid in every model. □

## 4.2 The Logical Consequence $\models$

The logical consequence pertinent to the issue of performability is semantically defined as a consequence relation  $\models$  operating on the set of all formulas of  $L$  in the following way. Let  $X$  be a set of formulas of  $L$  and  $\sigma$  a formula. We say that  $\sigma$  *logically follows* from  $X$ , in symbols:

$$X \models \sigma$$

if for every model  $M = \langle W, V, V_R \rangle$  and every state  $u \in W$ , if  $M, u \models \phi$  holds for all  $\phi \in X$ , then  $M, u \models \sigma$ . (The “big” symbol  $\models$  should not be confused with the “small” symbol  $\models$ , because they bear different meanings.)

It is clear that  $\models$  satisfies the standard conditions imposed on consequence relations (see e.g. Wójcicki 1988).

$\models$  validates the Deduction Theorem (DT): for any set  $X$  of formulas and any formulas  $\phi, \psi$ :

$$(DT) \quad X \models \phi \rightarrow \psi \text{ if and only if } X \cup \{\phi\} \models \psi.$$

Moreover  $\models$  validates the tautologies of classical logic expressed in  $L$  in the following sense. Let  $\phi(p_1, \dots, p_m)$  be a tautology of classical propositional logic and  $\sigma_1, \dots, \sigma_m$  atomic formulas of  $L$ . Then, the formula  $\phi(\sigma_1, \dots, \sigma_m)$  of  $L$  obtained from  $\phi(p_1, \dots, p_m)$  by the uniform substitution of  $\sigma_i$  for  $p_i, i = 1, \dots, m$ , is  $\models$ -valid. It is also easy to see that for any non-empty word  $x$  and any symbol  $a \in \Sigma$ , the formula  $Perf(xa) \rightarrow Perf(x)$  is a tautology of  $\models$ .

If  $A$  is a non-empty set of words, then for every  $x \in A$ , the formula

$$Perf(x) \rightarrow Perf(A)$$

is validated by the semantic consequence  $\models$ .

If  $A$  is a non-empty finite set of words,  $A = \{x_1, \dots, x_n\}$ , the formula

$$Perf(A) \leftrightarrow Perf(x_1) \vee \dots \vee Perf(x_n)$$

is  $\models$ -valid. Moreover for any subsets  $A, B$  of  $\Sigma^*$ , the formula

$$Perf(A \cup B) \leftrightarrow Perf(A) \vee Perf(B)$$

is also a tautology of  $\models$ . The formula  $\neg Perf(\emptyset)$  is also validated by  $\models$ .

A set  $X$  of formulas of  $L$  is *inconsistent* in the sense of  $\models$  if and only if  $X \models \phi$  for all formulas  $\phi$ , equivalently:  $X \models \phi \wedge \neg\phi$  for some (for all)  $\phi$ ; otherwise  $X$  is called *consistent*. A formula  $\sigma$  is *inconsistent* (resp. *consistent*) if the set  $\{\sigma\}$  is inconsistent (consistent).

It is easy to see that  $\sigma$  is  $\models$ -inconsistent if and only if  $M, u \models \sigma$  for no model  $M$  and no state  $u$  of  $M$ .

A set of formulas  $X$  is *closed* in the sense of  $\models$ , shortly:  $X$  is  $\models$ -closed, if  $X \models \sigma$  implies  $\sigma \in X$ , for every formula  $\sigma$ . The  $\models$ -closed sets are also called *theories of  $\models$* . They collectively form a closure system on  $L$ , denoted by  $Th(\models)$ .

### 5 Ultrasets

A *Lindenbaum set* of  $\models$  is a maximal consistent set  $\Delta$  in the sense of  $\models$ .

Every Lindenbaum set  $\Delta$  contains all instances of tautologies of classical propositional calculus (CPC) as well as specific action formulas. In particular,  $\Delta$  contains the formulas displayed at the end of the previous section. By maximality,  $\sigma \vee \tau \in \Delta$  if and only if  $\sigma \in \Delta$  or  $\tau \in \Delta$ , for all formulas  $\sigma, \tau$ . Moreover, also by maximality, each Lindenbaum set  $\Delta$  is  $\models$ -closed.

**Definition 5.1** A set  $\Delta$  of formulas of  $L$  is called an *ultra*set of  $\models$  if and only if  $\Delta$  is a Lindenbaum set of  $\models$  with the additional property holding for all sets  $A \subseteq \Sigma^*$ :

$$(ultra) \quad Perf(A) \in \Delta \Leftrightarrow Perf(x) \in \Delta \text{ for some word } x \in A. \quad \square$$

Since the implications  $Perf(x) \rightarrow Perf(A)$  are valid in the sense of  $\models$ , for all  $x \in A$ , we see that in the condition (ultra) only the implication ( $\Rightarrow$ ) matters.

Every ultraset, being maximal consistent, is logically closed.

The following fact follows from the above definition:

**Corollary 5.2** *A subset  $\Delta \subseteq \mathbf{L}$  is an ultraset for  $\models$  if and only if it is consistent and satisfies the following conditions for any formulas  $\sigma, \tau \in \mathbf{L}$  and any non-empty sets  $A \subseteq \Sigma^*$ :*

- (a)  $\sigma \wedge \tau \in \Delta \Leftrightarrow \sigma \in \Delta \text{ and } \tau \in \Delta.$
- (b)  $\neg\sigma \in \Delta \Leftrightarrow \sigma \notin \Delta,$
- (c)  $Perf(A) \in \Delta \Rightarrow Perf(x) \in \Delta \text{ for some word } x \in A.$
- (d)  $\neg Perf(\emptyset) \in \Delta.$  □

Ultrasets exist. This fact follows from the following observation:

**Proposition 5.3** *Let  $M = \langle W, V, V_R \rangle$  be a model and  $u$  a state in  $W$ . Define*

$$\Delta_u := \{\sigma \in \mathbf{L} : M, u \models \sigma\}.$$

*Then  $\Delta_u$  is an ultraset.*

**Proof** Straightforward. □

In fact, every Lindenbaum set of  $\models$  is an ultraset:

**Theorem 5.4** *For any set of formulas  $\Delta$ , the following conditions are equivalent:*

- (i)  $\Delta$  is a Lindenbaum set of  $\models$ ;
- (ii)  $\Delta$  is an ultraset set of  $\models$ ;
- (iii) *There is a model  $\langle W, V, V_R \rangle$  and a state  $u$  in  $W$  such that  $\Delta = \Delta_u$ .*

**Proof** The implications (iii)  $\Rightarrow$  (ii) and (ii)  $\Rightarrow$  (i) are obvious.

(i)  $\Rightarrow$  (iii). We need some facts from the theory of deductive systems.

By a *base* for  $Th(\models)$  we shall understand any family  $\mathbf{B} \subseteq Th(\models)$  such that every theory of  $\models$  is the intersection of some subfamily of  $\mathbf{B}$ . It follows from the definition of  $\models$  that the sets  $\Delta_u$  defined as in Proposition 5.3 (with  $u$  ranging over all states of arbitrary models  $M$ ) form a base for  $Th(\models)$ .

Now let  $\Delta$  be a Lindenbaum set of  $\models$ . Since  $\Delta$  is  $\models$ -consistent and closed, it is the intersection of a non-empty family of sets of the form  $\Delta_u$ . But inasmuch as  $\Delta$  is maximal,  $\Delta$  is equal to exactly one set of the form  $\Delta_u$ . This shows that (iii) holds. □

It follows from the above theorem that the family of ultrasets forms a basis for the closure system  $Th(\models)$  of all closed theories of  $\models$ , i.e., for every theory  $X \in Th(\models)$  there exists a family  $\{\Delta_i : i \in I\}$  of ultrasets such that  $X = \bigcap_{i \in I} \Delta_i$ . Another corollary is that ultrasets are the only maximal consistent sets of the consequence relation  $\models$ .

Each ultraset is fully determined by the set of atomic formulas of the form  $Perf(x)$ ,  $x \in \Sigma^*$ , it contains. This follows from the following fact:

**Proposition 5.5** *For any ultrasets  $\Delta$  and  $\Delta'$ ,*

$$(1) \Delta = \Delta' \Leftrightarrow (\forall x \in \Sigma^*)(Perf(x) \in \Delta \Leftrightarrow Perf(x) \in \Delta').$$

**Proof** The implication  $(\Rightarrow)$  is immediate. To prove the reverse implication  $(\Leftarrow)$ , assume RHS of (1). Then prove by induction on complexity of formulas that for any formula  $\sigma \in L, \sigma \in \Delta \Leftrightarrow \sigma \in \Delta'$ . (This proof requires Corollary 5.2)  $\square$

*Ultrasets* denotes the family of all ultrasets of  $\models$ .

The set *Ultrasets* is endowed with the following order relation  $\leq$ :

$$\Delta \leq \Delta' \Leftrightarrow_{df} (\forall x \in \Sigma^*)(Perf(x) \in \Delta \Rightarrow Perf(x) \in \Delta').$$

Let  $\Delta_0$  be the (unique) ulraset such that  $\neg Perf(x) \in \Delta_0$  for all words  $x \in \Sigma^*$ .  $\Delta_0$  is the least element in the poset  $\langle Ultrasets, \leq \rangle$  while the ulraset  $\Delta_1$  containing all formulas  $Perf(x), x \in \Sigma^*$  (including  $Perf(e)$ ) is the top element.

As each ulraset  $\Delta$  is unambiguously determined by the set  $\{x \in \Sigma^* : Perf(x) \in \Delta\}$ , we define the mapping  $h : Ultrasets \rightarrow \wp(\Sigma^*)$ :

$$h(\Delta) := \{x \in \Sigma^* : Perf(x) \in \Delta\}, \text{ for all } \Delta \in Ultrasets.$$

$h$  is an order isomorphism from the poset  $\langle Ultrasets, \leq \rangle$  onto the power set  $\langle \wp(\Sigma^*), \subseteq \rangle$ .

We also define, for each  $\Delta \in Ultrasets$ , the complement  $\Delta^c$  of  $\Delta$  to be the unique ulraset determined by the formulas  $Perf(x)$  such that  $\neg Perf(x) \in \Delta$ .

It follows that

$$h(\Delta^c) = \{x \in \Sigma^* : Perf(x) \in \Delta^c\} = \Sigma^* - h(\Delta),$$

for all  $\Delta \in Ultrasets$ .

$\Delta \rightsquigarrow \Delta^c$  is a well-defined unary operation on *Ultrasets*; it is marked by the symbol  ${}^c$ . We then get:

**Theorem 5.6** *The structure  $\langle Ultrasets, \leq, {}^c \rangle$  is a complete Boolean algebra isomorphic with the Boolean algebra of all subsets of  $\Sigma^*$  and the mapping  $h$  establishes this isomorphism.*  $\square$

If  $x$  and  $y$  are words in  $\Sigma^*$ , then the notation

$$y \preceq x$$

means that  $y$  is a prefix of  $x$ , i.e., there exists a word  $z$  such that  $yz = x$ .

We shall need some other properties of ultrasets. If  $y \preceq x$ , where  $y \neq e$ , and  $\Delta$  is an ulraset, then

$$Perf(x) \in \Delta \text{ implies that } Perf(y) \in \Delta.$$

This is due to the fact that  $Perf(x) \rightarrow Perf(y)$  is a tautology of  $\models$  whenever  $y \preceq x$  and  $y \neq e$ .

A function  $f : \Sigma^* \rightarrow \{0, 1\}$  is said to be *antitone* on  $\Sigma^*$  if  $f(x) = 1$  implies  $f(y) = 1$  whenever  $y \preceq x$  and  $y \neq e$ . In other words,  $f$  is antitone if  $f(y) = 0$  implies that  $f(x) = 0$  for all words  $x$  prefixed by a non-empty word  $y$ .

Let  $h$  be an antitone function defined on the set  $\Sigma^*$  of words with values in  $\{0, 1\}$ . For each word  $x$  we define the formula  $h(x) \text{ Perf}(x)$ :

$$h(x) \text{ Perf}(x) := \begin{cases} \text{Perf}(x) & \text{if } h(x) = 1 \\ \neg \text{Perf}(x) & \text{if } h(x) = 0. \end{cases}$$

Let  $H$  be the set of antitone functions on  $\Sigma^*$ . For each  $h \in H$ , we define

$$\Phi(h) := \{h(x) \text{ Perf}(x) : x \in \Sigma^*\}.$$

According to the above definition, for every formula of the form  $\text{Perf}(x)$ , either  $\text{Perf}(x) \in \Phi(h)$  or  $\neg \text{Perf}(x) \in \Phi(h)$ . Moreover, due to antitonicity of  $h$ ,  $\text{Perf}(x) \in \Phi(h)$  implies that  $\text{Perf}(y) \in \Phi(h)$  whenever  $y \preceq x$  and  $y \neq e$ .

The set  $\Phi(h)$  is called a *complete* set of atomic or negated atomic formulas. We claim that each set  $\Phi(h)$ , and therefore each function  $h \in H$ , determines a unique ultraset, denoted by  $\Delta(h)$ . By definition,  $\Delta(h)$  is the ultraset that contains the formulas of  $\Phi(h)$ .

## 5.1 Consistency of Complete Sets

The critical point is  $\models$ -consistency of  $\Phi(h)$ , for all  $h \in H$ . This subsection is devoted to this issue.

We shall show that for every  $h \in H$ , the set  $\Phi(h)$  is consistent in the sense of  $\models$ . To this end, we shall construct an appropriate model which will validate each set  $\Phi(h)$ . This will imply that the definition of the ultraset  $\Delta(h)$  is correct— $\Delta(h)$  is thus the unique ultraset that extends  $\Phi(h)$ , for all  $h$ .

Let

$$W =_{df} \{\Phi(h) : h \in H\}$$

be the family of *complete* sets of atomic or negated atomic formulas defined as above.

For each word  $x \in \Sigma^*$ , we define the total unary operation  $A(x)$  on the set  $W$  as follows. For each set  $\Phi \in W$ , we put:

$$(2) \quad A(x)(\Phi) := \{\text{Perf}(z) : z \in \Sigma^* \text{ and } \text{Perf}(xz) \in \Phi\}.$$

$A(x)$  is well-defined, because  $A(x)(\Phi) \in W$  whenever  $\Phi$  is in  $W$ . Thus each  $A(x)$  is a mapping from  $W$  to  $W$ .

In particular,

$$(3) \quad A(a)(\Phi) := \{\text{Perf}(z) : z \in \Sigma^* \text{ and } \text{Perf}(az) \in \Phi\},$$

for all  $a \in \Sigma$ .

It follows from the above definition and Proposition 5.5 that for the empty word  $e$ :

$$A(e) \text{ is the identity mapping in } W.$$

**Proposition 5.7** *For any words  $x, y \in \Sigma^*$ ,  $A(xy)$  coincides with the composition of the functions  $A(x)$  and  $A(y)$ , i.e.,*

$$A(xy) = A(x) \circ A(y).$$

**Proof.** We claim that for any  $\Phi \in W$

$$A(xy)(\Phi) = A(y)(A(x)(\Phi)).$$

We have

$$A(xy)(\Phi) := \{Perf(z) : z \in \Sigma^* \text{ and } Perf(xyz) \in \Phi\}.$$

As  $A(x)(\Phi) := \{Perf(u) : u \in \Sigma^* \text{ and } Perf(xu) \in \Phi\}$ , we obtain that

$$\begin{aligned} A(y)(A(x)(\Phi)) &= \{Perf(z) : z \in \Sigma^* \text{ and } Perf(yz) \in A(x)(\Phi)\} \\ &= \{Perf(z) : z \in \Sigma^* \text{ and } Perf(yz) \in \{Perf(w) : w \in \Sigma^* \text{ and } Perf(xw) \in \Phi\}\} \\ &= \{Perf(z) : z \in \Sigma^* \text{ and } Perf(xyz) \in \Phi\} = A(xy)(\Phi). \quad \square \end{aligned}$$

It follows from the above proposition that for any word  $x = a_1 \dots a_m$ , it is the case that

$$A(a_1 \dots a_m) = A(a_1) \circ \dots \circ A(a_m). \tag{1}$$

The unary ‘action’ operations  $A(x)$ ,  $x \in \Sigma^*$ , form a semigroup with respect of the operation of composition. Moreover the diagonal  $A(e)$  is the unit element of this semigroup. But we can say more:

**Theorem 5.8** *The mapping  $A$  is an isomorphism between the free semigroup  $(\Sigma^*, \circ, e)$  of words over  $\Sigma$  and the semigroup  $(\{A(x) : x \in \Sigma^*\}, \circ, A(e))$ .*

**Proof** As  $A$  is a surjective homomorphism, it suffices to show that it is one-to-one.

Let  $x$  and  $y$  be different words and let  $\Phi$  be a complete set such that the equivalence

$$(*) \text{ } Perf(x) \in \Phi \Leftrightarrow Perf(y) \in \Phi \text{ does not hold.}$$

Working with various possible configurations between two words  $x$  and  $y$ , one can find a set  $\Phi$  which validates (\*). E.g. if  $x$  is a proper prefix of  $y$ , i.e.,  $y = xu$  for some  $u \neq e$ , there is an complete set  $\Phi$  such that  $Perf(x) \notin \Phi$  and  $Perf(y) \in \Phi$ . The remaining cases can be similarly handled.

(\*) and the definitions of  $A(x)$  and  $A(y)$  imply that the equivalence

(\*\*)  $Perf(e) \in A(x)(\Phi) \Leftrightarrow Perf(e) \in A(y)(\Phi)$  does not hold.

Hence  $A(x) \neq A(y)$ . □

We set about defining a model  $M$  for  $L$ .

The set of states of  $M$  is equal to  $W$ , the family of all complete sets  $\{\Phi(h) : h \in H\}$ . We assume that the interpretation  $V(a)$  of each symbol  $a \in \Sigma$  is equal to the unary operation  $A(a)$  defined as above, i.e.,  $V(a) := A(a)$ .

All atomic actions  $V(a)$ ,  $a \in \Sigma$ , are therefore unary functions defined on  $W$ . Each function  $V(a)$  is extended onto arbitrary words  $x$  in accordance with Definition 3.3. Thus  $V(e)$  is the diagonal of  $W$  and  $V(xa) \stackrel{df}{=} V(x) \circ V(a)$ , for all words  $x \in \Sigma^*$  and all  $a \in \Sigma$ .

It follows from Proposition 5.7 that the mapping  $V$  coincides with  $A$ , i.e.,

$$V(x) = A(x),$$

for all words  $x \in \Sigma^*$ . In particular,  $V(e) = A(e)$ .

We then define partial functions  $V_R(a)$ ,  $a \in \Sigma$ . Each  $V_R(a)$  is the restriction of  $V(a)$  to the set  $\{\Phi \in W : Perf(a) \in \Phi\}$ . The last set is therefore the domain of  $V_R(a)$ . It should be noted that  $\{\Phi \in W : Perf(a) \in \Phi\}$  is a proper subset of  $W$ , for any  $a \in \Sigma$ . (This follows from the definition of complete sets.) Therefore  $V_R(a)$ , being a proper subset of  $V(a)$ , does not coincide with  $V(a)$ .

We also assume that  $V_R(e)$  is a partial function being the restriction of the identity mapping  $V(e)$  to the set  $\{\Phi \in W : Perf(e) \in \Phi\}$ .

Possible performances of  $V(a)$  are therefore ordered pairs  $\langle \Phi, V(a)(\Phi) \rangle$  with  $\Phi$  ranging over  $W$ . Not all possible performances  $\langle \Phi, V(a)(\Phi) \rangle$  of  $V(a)$  are realizable but only those for which  $Perf(a) \in \Phi$ .

We recursively extend the partial functions  $V_R(a)$  onto arbitrary *non-empty* words  $x$  in accordance with Definition 3.4. Thus

$$V_R(xa) := V_R(x) \circ V_R(a),$$

for all non-zero words  $x \in \Sigma^*$  and all  $a \in \Sigma$ .

It follows that

$$V_R(xy) = V_R(x) \circ V_R(y),$$

for all non-zero words  $x, y \in \Sigma^*$ .

**Theorem 5.9** *For any word  $x \in \Sigma^*$ ,  $V_R(x)$  is a partial function being the restriction of  $V(x)$  to the set  $\{\Phi \in W : Perf(x) \in \Phi\}$ .*

**Proof** We shall apply the graph-style notation for unary functions. The symbol ' $x F y$ ' means that ' $y = F(x)$ '.

The theorem will be proved once we show the following auxiliary lemma:

**Lemma** *For any word  $x \in \Sigma^*$  and any complete set  $\Phi$  the following conditions are equivalent:*

- (i)  $Perf(x) \in \Phi$ ,
- (ii) There exists a complete set  $\Phi'$  such that  $\Phi V_R(x)\Phi'$ .

**Proof** Fix a complete set  $\Phi$ . We shall prove the equivalence of (i) and (ii) by induction on the length of the words  $x$  of  $\Sigma^*$ .

**Induction base** In view of the definitions of  $V_R(e)$  and  $V_R(a)$ , the conditions (i) and (ii) are equivalent for the empty word  $e$  as well as for the symbols  $a$  of  $\Sigma$ .

**Induction step** We assume that (i) and (ii) are equivalent for a *non-empty* word  $x$  in  $\Sigma^*$ . We claim that this equivalence continues to hold for the word  $xa$ , for all  $a \in \Sigma$ .

We first assume (ii) holds for  $xa$ . Hence that there exist complete sets  $\Gamma, \Phi'$  such that

$$(a) \quad \Phi V_R(x)\Gamma V_R(a)\Phi'.$$

We want to show that  $Perf(xa) \in \Phi$ .

$\Gamma V_R(a)\Phi'$  in (a) means that

$$(b) \quad \Gamma V(a)\Phi' \text{ and } Perf(a) \in \Gamma.$$

$\Phi V_R(x)\Gamma$  in (a) implies that  $\Phi V(x)\Gamma$ . Hence, by the definition of  $V(x)$ ,

$$(\forall z \in \Sigma^*)(Perf(xz) \in \Phi \Leftrightarrow Perf(z) \in \Gamma).$$

Putting  $z = a$  and applying the second conjunct of (b), we obtain that  $Perf(xa) \in \Phi$ . So (i) holds.

Conversely, assume (i) holds, that is,  $Perf(xa) \in \Phi$ . We claim that there exists a complete set  $\Phi'$  such that  $\Phi V_R(xa)\Phi'$ .

If  $x = e$  we are done, by the definition of  $V_R(a)$ .

Assume that  $x$  is non-empty. The assumption  $Perf(xa) \in \Phi$  implies that  $Perf(x) \in \Phi$ , due to antinonicity of  $\Phi$ . As  $Perf(x) \in \Phi$ , IH gives that there exists a complete set  $\Gamma$  such that  $\Phi V_R(x)\Gamma$ . We claim that there exists a complete set  $\Phi'$  such that  $\Gamma V_R(a)\Phi'$ .

$$\Phi' := \{Perf(z) : Perf(az) \in \Gamma\} \cup \{\neg Perf(z) : \neg Perf(az) \in \Gamma\}.$$

$\Phi'$  is a complete set.

It follows from the definition of  $\Phi'$  that

$$(\forall z \in \Sigma^*)(Perf(az) \in \Gamma \Leftrightarrow Perf(z) \in \Phi').$$

So  $\Gamma V(a)\Phi'$  holds, i.e.,  $\Phi'$  is the value of  $V(a)$  at  $\Gamma$ . But we must also prove that  $\Gamma V_R(a)\Phi'$ , that is, we must show that  $Perf(a) \in \Gamma$ .

The proof that  $Perf(a) \in \Gamma$  runs as follows. Since  $\Phi V_R(x)\Gamma$ , we also have that  $\Phi V(x)\Gamma$ . As  $\Gamma V(a)\Phi'$ , we therefore obtain that  $\Phi V(xa)\Phi'$ . Applying Proposition 5.7 we therefore get that

$$(\forall z \in \Sigma^*)(Perf(xaz) \in \Phi \Leftrightarrow Perf(z) \in \Phi').$$

In particular, for  $z = e$ ,

(c)  $\text{Perf}(xa) \in \Phi \Leftrightarrow \text{Perf}(e) \in \Phi'$ .

But, by the assumption,  $\text{Perf}(xa) \in \Phi$ . It follows from (c) that  $\text{Perf}(e) \in \Phi'$ .

As  $\Gamma V(a)\Phi'$  holds, we have that

$$(\forall z \in \Sigma^*)(\text{Perf}(az) \in \Gamma \Leftrightarrow \text{Perf}(z) \in \Phi').$$

In particular, for  $z = e$ , we get that

$$\text{Perf}(a) \in \Gamma \Leftrightarrow \text{Perf}(e) \in \Phi'.$$

Since  $\text{Perf}(e) \in \Phi'$ , the above equivalence gives that  $\text{Perf}(a) \in \Gamma$ .

This concludes the proof that  $\Gamma V_R(a)\Phi'$  holds, showing at the same time that  $\Phi V_R(xa)\Phi'$ .

The proof of the lemma is completed.  $\square$

**Proposition 5.10** *Let  $\Phi$  be an arbitrary complete set in  $W$ . Then for every word  $x \in \Sigma^*$ ,*

$$M, \Phi \models \text{Perf}(x) \Leftrightarrow \text{Perf}(x) \in \Phi.$$

**Proof.** Suppose  $x \in \Sigma^*$ . We have:

$$M_c, \Phi \models \text{Perf}(x) \Leftrightarrow \text{(by Definition 3.1)}$$

$$\Phi \in \text{Perf } V_{seq}(x) \text{ in } M \Leftrightarrow \text{(by the definition of } \text{Perf } V_{seq}(x))$$

$$V_{seq}(x) \text{ is performable at } \Phi \text{ in } M \Leftrightarrow \text{(by Definition 3.5.(2))}$$

$$V_R(x) \text{ is defined at } \Phi \Leftrightarrow \text{(by the above lemma)}$$

$$\text{Perf}(x) \in \Phi. \quad \square$$

**Corollary 5.11** *For each  $h \in H$ , the complete set  $\Phi(h)$  is consistent.*

**Proof** Fix  $h \in H$ . In view of the above proposition, the set of all formulas  $\sigma$  of  $\mathbf{L}$  that are validated in the above model  $M$  at the state  $\Phi(h)$  encompasses all formulas of  $\Phi(h)$ . Hence  $\Phi(h)$  is consistent.  $\square$

Let  $\Delta(h)$  be the set of all formulas  $\sigma$  of  $\mathbf{L}$  validated at the state  $\Phi(h)$  in the above model;

$$\Delta(h) = \{\sigma \in \mathbf{L} : M_c, \Phi(h) \models \sigma\}.$$

In view of Proposition 5.3,  $\Delta(h)$  is an ultraset. In virtue of Proposition 5.10,  $\Delta(h)$  includes  $\Phi(h)$ . It follows from Proposition 5.5 that  $\Delta(h)$  is the unique ultraset that includes  $\Phi(h)$ .

It is also clear that for each ultraset  $\Delta$  there is a unique function  $h \in H$  such that  $\Delta = \Delta(h)$ .

As the set  $\Sigma^*$  is countably infinite, it follows that  $H$  is of cardinality of the continuum. Consequently, the cardinality of the family of ultrasets is equal to the continuum.

The definition of an ultraset does not imply that  $Perf(e)$  belongs to any ultraset. In fact,  $Ultrasets$  is the union of the following two disjoint sets:

$$\begin{aligned}
 Ultrasets^+ &:= \{\Delta \in Ultrasets : Perf(e) \in \Delta\}, \\
 Ultrasets^- &:= \{\Delta \in Ultrasets : \neg Perf(e) \in \Delta\}.
 \end{aligned}$$

We shall return to these sets later.

## 6 Operations on Ultrasets and the Canonical Model

In a fully analogous way to the definitions presented in the previous paragraph, we define for each word  $x \in \Sigma^*$ , the unary operation  $V(x)$  on the set  $Ultrasets$  as follows. For each  $\Delta \in Ultrasets$ , we put:

$$(1) \quad V(x)(\Delta) := \{Perf(z) : z \in \Sigma^* \text{ and } Perf(xz) \in \Delta\}.$$

$V(x)$  is well-defined, because  $V(x)(\Delta)$  is an ultraset whenever  $\Delta \in Ultrasets$ .  
 $V(e)$  is the identity map on the set  $Ultrasets$ .

Suitably modifying the proofs of the results presented in the previous paragraph we immediately obtain the following results analogous the the ones holding for the sets  $\Phi(h)$ ,  $h \in H$ .

**Proposition 6.1** *For any words  $x, y \in \Sigma^*$ ,  $V(xy)$  coincides with the composition of the functions  $V(x)$  and  $V(y)$ , i.e.,*

$$V(xy) = V(x) \circ V(y). \quad \square$$

**Theorem 6.2** *The mapping  $V$  is an isomorphism between the free semigroup  $(\Sigma^*, \circ, e)$  of words over  $\Sigma$  and the semigroup  $(\{V(x) : x \in \Sigma^*\}, \circ, V(e))$ .* □

### 6.1 The Canonical Model

We define a model  $M_c = \langle W, V, V_R \rangle$  for  $L$ .

The set of states  $W$  of  $M_c$  is equal to  $Ultrasets$ .

For each  $\Delta \in Ultrasets$  and  $a \in \Sigma$ ,  $V(a)$  is the operation defined as above, i.e.,

$$(2) \quad V(a)(\Delta) := \{Perf(z) : z \in \Sigma^* \text{ and } Perf(az) \in \Delta\}.$$

$V(e)$  is the identity map on the set  $Ultrasets$ .

$M_c$  is a deterministic model, because each  $V(a) : W \rightarrow W$  is a total function defined throughout  $W$ .

Each function  $V(a)$  is extended onto arbitrary words  $x$  in accordance with Definition 3.3. But it follows from Proposition 6.1 that the extension of  $V$  onto arbitrary word  $x$  coincides with the mapping  $V(x)$  defined according to formula (1) of Sect. 6.

In the next step we define the mappings  $V_R(a), a \in \Sigma$ . It is assumed that each  $V_R(a)$  is the partial function being the restriction of  $V(a)$  to the set  $\{\Delta \in W : Perf(a) \in \Delta\}$ . The last set is the domain of  $V_R(a)$ . It should be noted that  $\{\Delta \in W : Perf(a) \in \Delta\}$

is a proper subset of  $W$ , for all  $a \in \Sigma$ . Therefore the partial function  $V_R(a)$ , being a proper subset of  $V(a)$ , does not coincide with  $V(a)$ .

Speaking figuratively,  $V_R(a)$  is a counterpart of the green traffic light—once  $Perf(a)$  is in  $\Delta$ ,  $V(a)$  is performable at  $\Delta$ .

We also assume that  $V_R(e)$  is a partial function being the restriction of the identity mapping  $V(e)$  to the set  $\{\Delta \in W : Perf(e) \in \Delta\}$ .

The above model  $M_c$  is called the *canonical model* for  $L$ . Possible performances of  $V(a)$  are ordered pairs  $\langle \Delta, V(a)(\Delta) \rangle$  with  $\Delta$  ranging over *Ultrasets*. But according to the definition of  $V_R(a)$ , *realizable* performances of  $V(a)$  are ordered pairs  $\langle \Delta, V(a)(\Delta) \rangle$ , where  $Perf(a) \in \Delta$ , that is, all pairs  $\langle \Delta, V_R(a)(\Delta) \rangle$ .

The canonical model is isomorphic with the model defined in Sect. 5. The difference between the two models consists in the fact that the set of states of the former is equal to the set of ultrasets, while the sets of states of the latter coincides with the set of all complete sets.

The partial functions  $V_R(a)$  are recursively extended onto arbitrary non-empty words  $x$  in accordance with Definition 3.4. Thus

$$V_R(xa) := V_R(x) \circ V_R(a),$$

for all non-zero words  $x \in \Sigma^*$  and all  $a \in \Sigma$ .

It follows from the above definition that

$$V_R(xy) = V_R(x) \circ V_R(y),$$

for all *non-zero* words  $x, y \in \Sigma^*$ .

The following theorem is crucial. It is an analogue of Theorem 5.9:

**Theorem 6.3** *For any word  $x \in \Sigma^*$ ,  $V_R(x)$  is a partial function being the restriction of  $V(x)$  to the set  $\{\Delta \in W : Perf(x) \in \Delta\}$ .*

The proof is a straightforward modification of the proof of Theorem 5.9. □

Theorem 6.3 can be paraphrased in terms of action performability as follows:

**Theorem 5.3\*** *For any word  $x \in \Sigma^*$  and any ultraset  $\Delta$  the following conditions are equivalent:*

- (i)  $Perf(x) \in \Delta$ .
- (ii) *The sequential action  $V_{seq}(x)$  is performable at  $\Delta$ , i.e., there exists an ultraset  $\Delta'$  such that  $\Delta V_R(x) \Delta'$ .* □

Let  $\Delta$  be an ultraset such that  $Perf(a) \in \Delta$  for some symbol  $a$ . Then taking  $\Delta' := V(a)(\Delta)$ , we see that  $\Delta V_R(a) \Delta'$ . Thus  $\Delta$  is not a terminal state.

On the other hand, let  $\Delta_0$  be the ultraset defined as above (that is,  $\neg Perf(x) \in \Delta_0$  for all words  $x$ ). Then  $\Delta_0$  is a terminal state, that is,  $\Delta_0 V_R(x) \Delta'$  for no ultraset  $\Delta'$  and no word  $x$ . In fact,  $\Delta_0$  is the only terminal state and it is isolated in the following sense: there is no state  $\Delta$  and no word  $x$  such that  $\Delta V_R(x) \Delta_0$ . In other word, there is no way of reaching  $\Delta_0$  from any state. This follows from the following observation:

**Proposition 6.4** *For any ultraset  $\Delta$  and any word  $x$ ,  $Perf(e) \in V_R(x)(\Delta)$ .*

**Proof** Define  $\Delta' := \Delta V_R(x)$ . Then  $Perf(x) \in \Delta$ . It follows that

$$(\forall z \in \Sigma^*)(Perf(xz) \in \Delta \Leftrightarrow Perf(z) \in \Delta').$$

In particular, for  $z = e$ ,

$$Perf(x) \in \Delta \Leftrightarrow Perf(e) \in \Delta'.$$

As  $Perf(x) \in \Delta$ , the lemma follows. □

If  $Perf(e) \in \Delta$ , then obviously  $\Delta$  is a reflexive point of  $V_R(e)$ , that is,  $\Delta V_R(e)\Delta$ .

The following property of the canonical model is crucial:

**Lemma 6.5** (Truth Lemma) *Let  $\Delta$  be an arbitrary ultraset in the canonical model  $M_c$ . Then for any formula  $\phi$ ,*

(1)  $M_c, \Delta \models \phi$  if and only if  $\phi \in \Delta$ .

The proof is by induction on complexity of formulas. We first prove:

**Claim 1** For every word  $x \in \Sigma^*$ ,

$$M_c, \Delta \models Perf(x) \Leftrightarrow Perf(x) \in \Delta.$$

**Proof of the claim** Suppose  $x \in \Sigma^*$ . We have:

$$\begin{aligned} M_c, \Delta \models Perf(x) &\Leftrightarrow \text{(by Definition 4.1)} \\ \Delta \in \mathbf{Perf} V_{seq}(x) \text{ in } M_c &\Leftrightarrow \text{(by the definition of } \mathbf{Perf} V_{seq}(x)) \\ V_{seq}(x) \text{ is performable at } \Delta \text{ in } M_c &\Leftrightarrow \text{(by Definition 3.5)} \\ \Delta V_R(x)\Delta' \text{ for some state } \Delta' &\Leftrightarrow \text{(by Theorem 6.3*)} \\ Perf(x) \in \Delta. & \end{aligned}$$

This proves the claim. □

**Claim 2** For every set  $A \subseteq \Sigma^*$ ,

$$M_c, \Delta \models Perf(A) \Leftrightarrow Perf(A) \in \Delta.$$

**Proof of the claim** Suppose  $A$  is a subset of  $\Sigma^*$ . Then:

$$\begin{aligned} M_c, \Delta \models Perf(A) &\Leftrightarrow \\ \text{For some word } x \in A, M_c, \Delta \models Perf(x) &\Leftrightarrow \text{(by Claim 1)} \\ \text{For some word } x \in A, Perf(x) \in \Delta &\Leftrightarrow \\ Perf(A) \in \Delta. & \end{aligned}$$

(The last equivalence is due to the fact that  $\Delta$  is an ultraset). □

It follows from the above claims and Corollary 5.2 that the equivalence (1) continues to hold for arbitrary Boolean combinations of atomic formulas. This concludes the proof of the lemma. □

## 7 Other Properties of the Consequence Relation $\models$

The consequence relation determined by the canonical model  $M_c = \langle W, V, V_R \rangle$  on  $L$  agrees with  $\models$ . More specifically, we define the consequence relation  $\models_c$  on  $L$  as follows. For any set  $X \subseteq L$  and any formula  $\sigma \in L$  we put:

$$X \models_c \sigma \Leftrightarrow_{df} (\forall \Delta \in W)(M_c, \Delta \models X \Rightarrow M_c, \Delta \models \sigma).$$

(The symbol “ $M_c, \Delta \models X$ ” means that  $M_c, \Delta \models \phi$  holds for all  $\phi \in X$ .)  $\models_c$  is the consequence relation determined by  $M_c$ .

**Theorem 7.1** (*The Adequacy Theorem*)  $\models = \models_c$ .

**Proof** The inequality  $\models \leq \models_c$  is immediate, because  $\models$  is semantically defined by the class of all models that includes the canonical model.

To prove the opposite inequality, suppose that for some set  $X \subseteq L$  and a formula  $\sigma \in L$  it is *not* the case that  $X \models \sigma$ . We show that  $X \models_c \sigma$  does not hold.

According to the definition of  $\models$ , there exists then a model  $N = \langle W, V, V_R \rangle$  and a state  $u \in W$  such that  $N, u \models X$  and  $N, u \not\models \sigma$ . ( $N$  need not be the canonical model.) We then define:  $\Delta_u := \{\phi \in L : N, u \models \phi\}$ .  $\Delta_u$  is an ultraset,  $X \subseteq \Delta_u$  and  $\sigma \notin \Delta_u$ . Passing to the canonical model  $M_c$  we obtain, by the Truth Lemma, that  $M_c, \Delta_u \models X$  and  $M_c, \Delta_u \not\models \sigma$ . Consequently,  $X \models_c \sigma$  does not hold.  $\square$

The canonical model  $M_c = \langle W, V, V_R \rangle$  is deterministic: each  $V(a)$  is a total function on  $W$  and  $V_R(a)$  is a partial subfunction of  $V(a)$ . Hence  $V(a)$  is performable at  $u$  if and only if  $u$  is in the domain of  $V_R(a)$ .

The fact of adequacy of the canonical model for the logic of action  $\models$  has an impact on reach of the notion of performability. It follows from Theorem 7.1 that  $\models$  is fully determined by deterministic action models. Thus the logical strength of the notion of performability of atomic actions, encapsulated in Definition 3.5.(1), is reducible to performability of atomic actions in *deterministic* action models. The phenomenon of unperformability of an atomic action  $a$  in a given state  $u$  in a deterministic model is tantamount to the fact that  $u$  is not in the domain of the partial subfunction  $V_R(a)$  of the unary function  $V(a)$ —this action simply halts at some state.

As to performability of actions in situational action systems—the problem is more involved there (see Czelakowski 2015).

We shall establish some other facts concerning  $\models$ .

**Proposition 7.2** *If  $\Sigma$  is an infinite set, the consequence  $\models$  is not finitary.*

**Proof** Assume  $\Sigma = \{a_n : n \in \mathbb{N}\}$ . We put  $A := \Sigma$ .  $\square$

**Claim** (1)  $\{\neg \text{Perf}(a_n) : n \in \mathbb{N}\} \models \neg \text{Perf}(A)$ ;

(2) *For every finite subset  $\mathbb{N}_f \subset \mathbb{N}$ , it is not the case that*

$$\{\neg \text{Perf}(a_n) : n \in \mathbb{N}_f\} \models \neg \text{Perf}(A).$$

**Proof of the claim** As to the first statement, suppose  $M = \langle W, V, V_R \rangle$  is a model for  $L$  and  $u \in W$  is a state such that  $M, u \models \neg Perf(a_n)$  for all  $n \in \mathbb{N}$ . This means that  $u \notin Perf V(a_n)$  for all  $n \in \mathbb{N}$  in  $M$ . It follows that  $u \notin Perf V(A)$ . Hence  $M, u \models \neg Perf(A)$ .

To prove the other statement, it suffices to show that for every positive integer  $m$  it is not the case that  $\{\neg Perf(a_n) : n \leq m\} \models \neg Perf(A)$ . To this end we take a frame-based normalized model  $M = \langle W, R, V \rangle$  and a state  $u \in W$  with the property that for each  $a_n, n \leq m$ , there is no state  $v$  such that  $uV(a_n)v$  but there exists a state  $w$  for which  $uV(a_{m+1})w$  holds. Such a model  $M$  can be easily defined. It follows that each action  $V(a_n), n \leq m$ , is unperformable at  $u$  but  $V(a_{m+1})$  is performable at  $u$ . Consequently,  $M, u \models \neg Perf(a_n)$  for all  $n \leq m$ . On the other hand,  $M, u \models Perf(A)$ , because  $M, u \models Perf(a_{m+1})$ . Thus  $\{\neg Perf(a_n) : n \leq m\} \models \neg Perf(A)$  does not hold.

This proves the claim and concludes the proof of the proposition. □

But we can say even more:

**Theorem 7.3** *If  $\Sigma$  has at least two elements, then the consequence  $\models$  is not finitary.*

**Proof** We shall suitably accommodate the Proof of Proposition 7.2 to the above assumption. Let  $a$  and  $b$  be different symbols in  $\Sigma$ . We define:

$$A := \{ab^n a : n \geq 1\}.$$

□

- Claim** (1)  $\{\neg Perf(ab^n a) : n \in \mathbb{N}\} \models \neg Perf(A)$ ;  
 (2) For every finite subset  $\mathbb{N}_f \subset \mathbb{N}$ , it is not the case that

$$\{\neg Perf(ab^n a) : n \in \mathbb{N}_f\} \models \neg Perf(A).$$

**Proof of the claim** As to the first statement, suppose  $M = \langle W, V, V_R \rangle$  is a model for  $L$  and  $u \in W$  is a state such that  $M, u \models \neg Perf(ab^n a)$  for all  $n \in \mathbb{N}$ . This means that  $u \notin Perf V_{seq}(ab^n a)$  for all  $n \in \mathbb{N}$  in  $M$ . As  $Perf V_{seq}(A) = \bigcup_{n \in \mathbb{N}} Perf V_{seq}(ab^n a)$ , it follows that  $u \notin Perf V_{seq}(A)$ . Hence  $M, u \models \neg Perf(A)$ .

To prove the other statement, it suffices to show that for every positive integer  $m$  it is not the case that  $\{\neg Perf(ab^n a) : n \leq m\} \models \neg Perf(A)$ . To this end we take a normalized frame-based model  $M = \langle W, R, V \rangle$ , a state  $u \in W$  and a sequence of different states  $u_0, u_1, \dots, u_m, u_{m+1}$  and  $w_{m+1}$  such that  $uRu_0Ru_1R \dots u_mRu_{m+1}Rw_{m+1}$  is the only terminated  $R$ -path that originates from  $u$ . Moreover, it is assumed that

$$uV(a)u_0V(b)u_1V(b)u_2 \dots u_mV(b)u_{m+1}V(a)w_{m+1}$$

and for each  $n, 1 \leq n \leq m$ , there is no state  $w_n$  such that  $u_nV(a)w_n$ .

Such a model  $M$  can be easily defined. It follows that the action  $V(a)$  is unperformable at each state  $u_n, 1 \leq n \leq m$ , but it is performable at  $u_{m+1}$ . Consequently, by the uniqueness of the above sequence of states, we have that  $M, u \models \neg Perf(ab^n a)$  for

all  $n \leq m$ . On the other hand,  $M, u \models Perf(A)$ , because  $M, u \models Perf(ab_{n+1}a)$ . Thus it is not the case that  $\{\neg Perf(ab^n a) : n \leq m\} \models \neg Perf(A)$ .

This proves the claim and concludes the Proof of the Theorem.  $\square$

It is clear that the unbounded “pumping” the symbol  $b$  in the words of the set  $A$  accounts for the fact that  $\models$  is not finitary. Such “pumping” features of formal languages are well known in automata theory. They result in various  $\omega$ -type rules of inference as e.g. the rule (1) from the above claim.

The following fact sharply contrasts with the above one:

**Theorem 7.4** *If  $\Sigma$  has one element only, then the consequence  $\models$  is finitary.*

**Proof** Let us assume that  $\Sigma = \{a\}$ . The proof of the theorem is based on the following lemma:

**Lemma** *If  $\Sigma$  is a singleton, then for each non-empty set  $A \subseteq \Sigma^*$  not containing the empty word  $e$  there is a word  $x \in A$  such that  $\models Perf(A) \leftrightarrow Perf(x)$ .*

**Proof of the lemma** Let  $A$  be a non-empty subset of  $\Sigma^*$ . Let  $x = a^n$  be the shortest word in  $A$ .  $x$  is non-empty. As  $x \in A$ , we have that  $\models Perf(x) \rightarrow Perf(A)$ . We claim that  $\models Perf(A) \rightarrow Perf(x)$ .

Let  $M = \langle W, V, V_R \rangle$  be a model for  $L$  and  $u \in W$  a state such that  $M, u \models Perf(A)$ . Then  $M, u \models Perf(y)$  for some word  $y \in A$ . But by the definition of  $x$ , the word  $x$  is a prefix of  $y$ . As  $M, u \models Perf(y)$ , we also have that  $M, u \models Perf(x)$ , because performability of  $V(y)$  at  $u$  entails performability of  $V(z)$  at  $u$  for all non-zero prefixes  $z$  of  $y$ . Consequently,  $M, u \models Perf(x)$ .  $\square$

The above lemma states that all formulas of the form  $Perf(A)$  are eliminable on the basis of  $\models$ , because each such a formula is  $\models$ -equivalent to a formula of the form  $Perf(x)$ . Then a straightforward argument shows that  $\models$  is finitary. Details are left to the reader.  $\square$

## 7.1 A Finitary Strengthening of the Consequence $\models$

The question arises if there are finitary and non-trivial strengthenings of the above consequence  $\models$ . We shall shed some light on this issue.

Let  $\Sigma$  be an alphabet. The infinitary consequence  $\models$ , defined in Sect. 4, is determined by *all* action models over  $\Sigma$ . We define a narrower class of models. Let  $\mathbf{K}$  be the class of action models  $M = \langle W, V, V_R \rangle$  over  $\Sigma$  such that, for every  $a \in \Sigma$ ,  $V(a)$  is serial relation on the set of states  $W$  and moreover  $V_R(a) = V(a)$ , for all  $a$ . Thus every atomic action  $V(a)$  is performable in all states  $u \in W$ .

It is easy to see that the class  $\mathbf{K}$  is first-order definable and therefore it is closed under the formation of ultraproducts.

$\models_{\mathbf{K}}$  is the consequence relation in the language  $L$  determined by the models of  $\mathbf{K}$ .  $\models_{\mathbf{K}}$  is stronger than  $\models$ .

All formulas  $Perf(a)$ ,  $a \in \Sigma$ , are tautologies of the consequence  $\models_{\mathbf{K}}$ . It follows that every formula  $Perf(x)$ ,  $x \in \Sigma^*$ , is also a tautology of  $\models_{\mathbf{K}}$ . Consequently, for any

non-empty compound action  $A \subseteq \Sigma^*$ , the formula  $Perf(A)$  is a tautology of  $\models_{\mathbf{K}}$  as well. In other words, in any model  $M \in \mathbf{K}$ , the action  $V_{seq}(A)$  is performable in all states of  $M$ .

In fact, for any non-empty  $A \subseteq \Sigma^*$ ,  $Perf(x) \rightarrow Perf(A)$  and  $Perf(A) \rightarrow Perf(x)$  are tautologies of  $\models_{\mathbf{K}}$ , for all  $x \in A$ . Thus the logic  $\models_{\mathbf{K}}$  trivializes the problem of performability.

If  $M_i = \langle W_i, V_i, V_{i,R} \rangle$ ,  $i \in I$ , is an indexed, nonempty family of models in  $\mathbf{K}$ , and  $U$  is an ultrafilter over  $I$ , then in the standard way one defines the ultraproduct  $M = \Pi_U M_i$  of this family modulo  $U$ . The model  $M$  also belongs to  $\mathbf{K}$ .

We have the following counterpart of Łoś’s Theorem for the formulas of  $\mathbf{L}$  in ultraproducts of models of  $\mathbf{K}$ :

*Let  $M = \Pi_U M_i$  be an ultraproduct of a family  $M_i = \langle W_i, V_i, V_{i,R} \rangle$ ,  $i \in I$ , of  $\mathbf{K}$ -models and let  $w_i \in W_i$  be a state,  $i \in I$ . Then for any formula  $\phi \in \mathbf{L}$ ,*  
 (\*)  $M, \langle w_i : i \in I \rangle_U \models \phi \Leftrightarrow \{i \in I : M_i, w_i \models \phi\} \in U$ .

(\*) is recursively proved by the degree of complexity of  $\phi$ . The only non-trivial case is when  $\phi$  is of the form  $Perf(A)$ , where  $A$  is a non-empty, possibly infinite compound action. We apply the fact that  $Perf(A)$  is a tautology of  $\models_{\mathbf{K}}$ . Hence  $M, \langle w_i : i \in I \rangle_U \models Perf(A)$  and  $M_i, w_i \models Perf(A)$  for all  $i \in I$ , because all these models are in  $\mathbf{K}$ . Hence  $\{i \in I : M_i, w_i \models Perf(A)\} \in U$ . Thus both sides of (\*) are true for  $\phi$  equal to  $Perf(A)$  and (\*) follows.

By applying (\*) and a standard ultraproduct argument one proves:

**Theorem 7.5** *The consequence relation  $\models_{\mathbf{K}}$  is finitary.*

Here is a brief outline of the proof. Assume  $X \subseteq \mathbf{L}$ ,  $\phi \in \mathbf{L}$  and  $X \models_{\mathbf{K}} \phi$ . We claim that  $X_f \models_{\mathbf{K}} \phi$  for some finite set  $X_f \subseteq X$ . We argue by contraposition. Let  $I$  be the family of all finite non-empty subsets of  $X$ . Suppose that  $i \not\models_{\mathbf{K}} \phi$  for all  $i \in I$ . For each  $i$  we find a model  $M_i = \langle W_i, V_i, V_{i,R} \rangle$  in  $\mathbf{K}$  and a state  $w_i \in W_i$  such that  $M_i, w_i \models i$  (i.e., all formulas  $\alpha \in i$  are validated by  $w_i$  in  $M_i$ ) but  $\phi$  is not, i.e.,  $M_i, w_i \not\models \phi$ . Let  $U$  be an ultrafilter over  $I$  which includes the sets  $i^* := \{j \in I : i \subseteq j\}$ ,  $i \in I$ , and let  $M = \Pi_U M_i$  be the ultraproduct of the models  $M_i$ ,  $i \in I$ , modulo  $U$ . Then by (\*), the state  $w := \langle w_i : i \in I \rangle_U$  belonging to  $\Pi_U W_i$  does validates all formulas  $X$  but it does not validate  $\phi$ . As  $M$  is in  $\mathbf{K}$ , it follows that  $X \not\models_{\mathbf{K}} \phi$ . Hence the consequence  $\models_{\mathbf{K}}$  is finitary. □

By using a similar argument one can define other finitary consequence relations of actions.

### 8 The Logic of Finite Actions

The fact that the consequence operation  $\models$  is not finitary poses the problem of its finitarization. In metalogic one sometimes applies various methods of finitarization to infinitistic propositional calculi to obtain a finitary version of the calculi. The infinite-valued Łukasiewicz consequence defined by the matrix  $(A_c, \{1\})$  may serve as an example—see the last section below. The consequence determined by this matrix is infinitistic; but in the theory of Łukasiewicz logics one usually works with the finitary

version of this consequence. At the semantic level, one may also apply techniques based on ultraproducts to yield finitary consequences. But these methods are not applicable to  $\models$ .

In this section the discourse on performability of compound actions is restricted to the special case when all compound formal actions in question are finite. They thus form a countably infinite subfamily of the power set of  $\Sigma^*$ .

We define the sublanguage  $L_f$  of  $L$  as follows. Atomic formulas of  $L_f$  are expressions of the form:

$$\begin{aligned} \text{Perf}(x), \quad x \in \Sigma^*, \\ \text{Perf}(A), \quad A \text{ is a finite subset of } \Sigma^*. \end{aligned}$$

As the set  $\Sigma^*$  is countably infinite, the above set of atomic formulas is countably infinite as well.

Compound formulas are built from the above atomic formulas by means of applying the Boolean connectives  $\rightarrow$  and  $\neg$ . The other Boolean connectives are defined in the standard way.

Models for  $L_f$  are defined as for the language  $L$ . Satisfaction in models is also defined as for  $L$ .

$\models_f$  is the semantic consequence relation in  $L_f$  defined in the same way as for  $\models$ . It follows that  $\models_f$  is the restriction of  $\models$  to  $L_f$ . (Note that the symbol  $\models_f$  has here *different* meaning from that in the above section.)

## 8.1 The Logic $\vdash$

We want to syntactically characterize the above semantically defined consequence relation  $\models_f$  in terms of logical axioms and rules of inference. (In fact, only one primitive rule of inference is needed here—the rule of detachment.) To this end we first define an inferentially defined consequence relation in  $L_f$ , denoted by  $\vdash$ . The consequence  $\vdash$  is an extension of classical propositional logic (CPC).

Every formula of  $L_f$  which is an instance of a tautology of CPC is logically valid. But there are also logically valid formulas specific to action performability.

As an axiom system of classical logic we adopt the following laws:

$$\begin{aligned} \text{(a}_1\text{)} \quad \phi \rightarrow (\psi \rightarrow \phi) & \qquad \text{(simplification)} \\ \text{(a}_2\text{)} \quad \phi \rightarrow (\psi \rightarrow \chi) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \chi)) & \quad \text{(Frege's Syllogism)} \\ \text{(a}_3\text{)} \quad \neg\phi \rightarrow (\phi \rightarrow \psi) & \qquad \text{(Duns Scotus' Law)} \\ \text{(a}_4\text{)} \quad (\neg\phi \rightarrow \phi) \rightarrow \phi & \qquad \text{(Clavius' Law)} \end{aligned}$$

where  $\phi, \psi, \chi$  are arbitrary formulas.

We have the following specific action axioms:

$$\text{(d}_1\text{)} \quad \text{Perf}(xa) \rightarrow \text{Perf}(x),$$

where  $x$  is a non-empty word and  $a \in \Sigma$ ;

$$(d_2) \quad \neg Perf(\emptyset).$$

Moreover, for each non-empty finite set  $A = \{x_1, \dots, x_n\} \subset \Sigma^*$  we adopt:

$$(d_3) \quad Perf(A) \leftrightarrow Perf(x_1) \vee \dots \vee Perf(x_n).$$

The detachment rule given by the scheme  $\phi, \phi \rightarrow \psi / \psi$  is the only primitive rule of inference.

We define

$$\vdash$$

to be the consequence relation in  $L$  determined by the above specific action axioms (d<sub>1</sub>)–(d<sub>3</sub>), the above axiom system (a<sub>1</sub>)–(a<sub>4</sub>) for CPC and the detachment rule. Thus  $X \vdash \sigma$  means that there is proof of  $\sigma$  from  $X$  carried out by means of the above logical axioms and the detachment rule.  $\vdash$  is called the *inferential performability consequence* in  $L$ .

$\vdash$  is finitary. Since  $\vdash$  is based on classical logic and the detachment as the only primitive rule of inference,  $\vdash$  obeys the Deduction Theorem, which means that for any set  $X$  of formulas and any formulas  $\phi, \psi$ :

$$X \vdash \phi \rightarrow \psi \quad \text{if and only if} \quad X \cup \{\phi\} \vdash \psi.$$

The axioms (d<sub>3</sub>) *de facto* assure eliminability of atomic formulas of the form  $Perf(A)$  from  $L_f$ , because each such formula with  $A = \{x_1, \dots, x_n\}$  can be replaced by the deductively equivalent disjunction  $Perf(x_1) \vee \dots \vee Perf(x_n)$ .

For every finite non-empty set  $A$  the formula  $Perf(x) \rightarrow Perf(A)$  is a thesis of  $\vdash$ , for all words  $x \in A$ . Moreover, for any non-empty words  $x, y$ , the formula  $Perf(xy) \rightarrow Perf(x)$  is a thesis too. This can be shown by iterating the axioms (d<sub>1</sub>).

We also note that for any finite subsets  $A, B$  of  $\Sigma^*$ , the formula

$$Perf(A \cup B) \leftrightarrow Perf(A) \vee Perf(B).$$

is a thesis of  $\vdash$ .

A set  $X$  is *inconsistent* if all formulas are  $\vdash$ -consequences of  $X$ , equivalently, if a formula of the form  $\phi \wedge \neg\phi$  is derivable from  $X$ ; otherwise  $X$  is *consistent*. A formula  $\sigma$  is *inconsistent* if the set  $\{\sigma\}$  is inconsistent. Analogously one defines consistency of a formula.

Since the above axioms are validated in all models, we see that  $\vdash \leq \models_f$ , i.e.,  $\models_f$  is stronger than  $\vdash$ .

$\vdash$  is a variant of CPC. Therefore the (algebraic) closure system  $Th(\vdash)$  has a base consisting of all Lindenbaum sets of  $\vdash$ , i.e., maximal consistent subsets of  $L_f$ .

Due to the axioms (d<sub>3</sub>), each Lindenbaum set  $\Delta$  is an ultraset in the sense that for any non-empty finite set  $A \subset \Sigma^*$ ,  $Perf(A) \in \Delta$  implies that  $Perf(x) \in \Delta$  for some  $x \in A$ . Moreover, for any non-empty words  $x$  and  $y$ ,  $Perf(xy) \in \Delta$  implies  $Perf(x) \in \Delta$ .

Every Lindenbaum set of  $\vdash$  is therefore characterized as in Corollary 5.2 (under the condition that  $A$  ranges over finite sets). The  $\vdash$ -counterparts of Proposition 5.3 and Theorem 5.4 also hold (but restricted to the language  $L_f$ ).

The models for  $\vdash$  as well as the canonical model  $M_c = \langle W, V, V_R \rangle$  for  $\vdash$  are defined in a fully analogous way as for  $\models$ . (The only difference is that the system  $\vdash$  is syntactically defined by means of logical axioms and the detachment rule, and not through models. But the axioms of  $\vdash$  enable us to define all components of  $M_c$  in the same way as in the case of the system  $\models$ .)

As to the canonical model, its set of states  $W$  is equal to the family of Lindenbaum sets of  $\vdash$ , and hence ultraset in the above sense. The unary functions  $V(a)$ ,  $a \in \Sigma$ , acting on the set  $W$  are defined as in Sect. 6. In an analogous way one defines the partial functions  $V_R(a)$ ,  $a \in \Sigma$ , and  $V_R(e)$ . Proposition 6.1 and Theorem 6.3 then continue to hold for Lindenbaum sets of the system  $\vdash$ . The interpretation  $V$  is defined as Sect. 6, that is,  $V(a) := A(a)$ ,  $a \in \Sigma$ , and  $V(e) := A(e) =$  the identity map on  $W$ .

Lemma 6.5 also continues to hold for the above canonical model (without significant modifications in its proof; the only restriction is that  $A$  ranges over finite subsets of  $\Sigma^*$ ):

**Lemma 8.1** (Truth Lemma) *Let  $\Delta$  be an arbitrary Lindenbaum set in the above canonical model  $M_c$ . Then for any formula  $\phi$  of  $L_f$ :*

(1)  $M_c, \Delta \models \phi$  if and only if  $\phi \in \Delta$ . □

The following fact is the main result of this part of the paper.

**Theorem 8.2** (The Extended Completeness Theorem)  $\vdash = \models_f$ .

**Proof** The inclusion  $\vdash \subseteq \models_f$  is immediate, because the axioms of  $\vdash$  are logically valid and *Modus Ponens* is a rule of  $\models_f$ .

To prove the reverse inclusion, let us assume that  $X$  is a set of formulas of  $L_f$  and  $\sigma$  is a formula such that it is *not* the case that  $X \vdash \sigma$ . We shall show that  $\sigma$  does not logically follow from  $X$ .

There is a Lindenbaum set  $\Delta_0$  of  $\vdash$  such that  $X \subseteq \Delta_0$  and  $\sigma \notin \Delta_0$ . Let  $M_c = \langle W, V, V_R \rangle$  be the canonical model for  $L_f$  defined as above. Hence  $\Delta_0 \in W$ .

As  $X \subseteq \Delta_0$  and  $\sigma \notin \Delta_0$  we obtain, by Lemma 8.1, that every formula of  $X$  is true in  $M_c$  at  $\Delta_0$ . Since  $\sigma \notin \Delta_0$ , we have that  $\sigma$  is not true in  $M_c$  at  $\Delta_0$ . Consequently, it is not the case that  $X \models_f \sigma$ .

This shows the inclusion  $\models_f \subseteq \vdash$ . □

It follows from the above theorems that the finitary consequence relation  $\vdash$  is semantically defined by deterministic models.<sup>2</sup>

We note that for any finite subsets  $A, B$  of  $\Sigma^*$ , the formula

$$Perf(A \cup B) \leftrightarrow Perf(A) \vee Perf(B)$$

is a thesis of  $\vdash$ , because it is validated by  $\models_f$ .

<sup>2</sup> This resembles a situation in automata theory where all regular languages are determined by finite deterministic automata.

### 9 Regular Actions: Comparisons with Dynamic Logic

To enlarge the expressive power of  $L$ , one may extend the above vocabulary of  $L$  by incorporating propositional variables. In fact, one may also make another step and define more nuanced forms of performability of actions by combining the propositions  $Perf(x)$ ,  $x \in \Sigma^*$ , and  $Perf(A)$ ,  $A \subseteq \Sigma^*$ , with propositional variables. These forms reflect *local* aspects of action performability, not on the entire set of states but on its subsets. Accordingly, for any propositional variable  $p$ , we define the formulas

$$Perf(x) \wedge p \quad \text{and} \quad Perf(A) \wedge p.$$

They are read: “ $x$  is performable in  $p$ ” and “ $A$  is performable in  $p$ ”, respectively. In the semantic rendition, if  $M = \langle W, V, V_R \rangle$  is an action model, and  $\Phi \subseteq W$  is the interpretation of  $p$ , then  $Perf(x) \wedge p$  is interpreted as

$$\Phi \cap Perf \ V_{seq}(x) := \{u \in \Phi : (\exists w \in W) uV_R(x)w\}.$$

Analogously,  $Perf(A) \wedge p$  is interpreted as

$$\Phi \cap Perf \ V_{seq}(A) := \{u \in \Phi : (\exists x \in A)(\exists w \in W) uV_R(x)w\}.$$

But, more interestingly, we may also define local counterparts of the “reversed performability” (see Notes 2.7.B) as the following unary operations:

$$\begin{aligned} w \in Perf \leftarrow V_{seq}(x)(\Phi) &\Leftrightarrow_{df} (\exists u \in \Phi) uV_R(x)w, \\ w \in Perf \leftarrow V_{seq}(A)(\Phi) &\Leftrightarrow_{df} (\exists x \in A) w \in Perf \leftarrow V_{seq}(x)(\Phi), \end{aligned}$$

for all subsets  $\Phi \subseteq W$ .

Thus  $Perf \leftarrow V_{seq}(x)(\Phi)$  is the set of states attainable from  $\Phi$  by realizable performances of  $V_{seq}(x)$ . Analogously one can read the proposition  $Perf \leftarrow V_{seq}(A)(\Phi)$ .

The above issues, though important, are not discussed in this paper.

Dynamic logic (DL) is concerned with the logic of action but it perceives actions from a different perspective than the one articulated in this work. DL steams from the theory of programs. It is well known that “DL is an extension of modal logic originally intended for reasoning about computer programs and later applied to more general complex behaviors arising in linguistics, philosophy, AI, and other fields” (after Wikipedia). Although any action variable  $a$  is interpreted in DL as a binary relation on a set of states, thus as an atomic action, DL examines the actions corresponding to action variables from the modal logic perspective (see e.g. Parikh (1978), Segerberg (1980, 1982)). The action variables are viewed there as modal operator makers. More specifically,  $a$  gives rise to the modal connective  $[a]$ , which, in the syntactic rendition, is a kind of the modal necessity connective. If  $p$  is a sentential variable, then  $[a]p$  is read as follows: “After performing the action  $a$  it is necessarily the case that  $p$ ”, that is,  $a$  must bring about  $p$ . The dual modal connective  $\langle a \rangle$  is interpreted as a possibility connective.  $\langle a \rangle p$  is therefore read: “After performing  $a$ , it is possible that  $p$  holds”, that is,  $a$  might bring about  $p$ . DL also contains more complicated modal connectives

that correspond to regular compound actions. DL abstracts from the issue of whether the action represented by  $a$  is performable in a given circumstances or not, but is interested in results the action brings about *whenever it is performed*.

Consequently, DL assumes a quite different linguistic perspective of action as compared with the one presented in this work. The sentential language of DL is a multimodal language. This language is an extension of the standard Boolean language of CPC. In turn, the language  $L$  of action defined in this paper has an entirely different syntax. There are no sentential variables at all. Atomic formulas are of the form  $Perf(x)$ , where  $x$  is a word over an alphabet  $\Sigma$ , or of the form  $Perf(A)$ , where  $A$  is a subset of  $\Sigma^*$ .

One may expand the language  $L$  of action performability and add propositional variables as well as modal connectives  $[a]$ ,  $a \in \Sigma$ , to the vocabulary of  $L$  together with other specific connectives of DL. This fact, due to the presence of Boolean connectives in  $L$ , enables one to incorporate the language of dynamic logic to  $L$ . But one may even go farther and augment the language of  $L$  with the modal necessity connectives  $[x]$ ,  $x \in \Sigma^*$ , and  $[A]$ ,  $A \subseteq \Sigma^*$ . There are altogether uncountably many such necessity connectives. (But of course, the set  $\{[x] : x \in \Sigma^*\}$  is countably infinite.)

$[x]p$  is read as follows: “After performing the sequence of actions  $x$  it is necessarily the case that  $p$ ” while  $[A]p$  is read “After performing the compound action  $A$  it is necessarily the case that  $p$ ”.

Although from the purely formal viewpoint, it is easy to combine the syntax of  $L$  with the syntax of dynamic logic, as described above, as well as to extend the relational semantics of  $L$  so that it captures the semantic principles of DL, some serious logical problems remain. Dynamic logic, being a variant of multimodal logic, is finitary. DL is axiomatically defined. The completeness theorem for DL with respect to the relational semantics is crucial. DL is even an equivalential logic in the well-known hierarchy of protoalgebraic logics—see e.g. (Czelakowski 2001, §3.6). In turn, the logic of performability is inherently infinitary, as this paper shows. It is easy to see that the logic which combines the logic of performability with DL is infinitary as well. And this infinitistic aspect of such an extended logic is not easily removable in general.

The family  $REG(\Sigma)$  of regular languages over  $\Sigma$  possesses the well known algebraic properties. The purpose of this section is to present a modified definition of performability of compound actions as compared with the options Definition 3.5.(3) offers. This new definition is strictly tailored for regular actions and only for them. The definition of performability of regular actions we shall present directly refers to the algebraic structure of  $REG(\Sigma)$  and it takes into account this structure.

Given an action model  $M = \langle W, V, V_R \rangle$  over  $\Sigma$  and  $a \in \Sigma$  we define the unary modal operations  $\langle V_R(a) \rangle$  and  $[V_R(a)]$  on the subsets of  $W$  as follows. For  $\Phi \subseteq W$  and  $u \in W$ ,

- (1)  $u \in \langle V(a) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV(a)w \wedge w \in \Phi).$
- (2)  $u \in [V(a)]\Phi \Leftrightarrow_{df} (\forall w)(uV(a)w \Rightarrow w \in \Phi).$
- (3)  $u \in \langle V_R(a) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV_R(a)w \wedge w \in \Phi).$
- (4)  $u \in [V_R(a)]\Phi \Leftrightarrow_{df} (\forall w)(uV_R(a)w \Rightarrow w \in \Phi).$

$\langle V(a) \rangle \Phi$  is the  $V$ -preimage of the set  $\Phi$  and  $\langle V_R(a) \rangle \Phi$  is the  $V_R$ -preimage of  $\Phi$ . It is clear that the operations  $\langle V(a) \rangle$  and  $[V(a)]$  are mutually interdefinable in the well-known way and similarly the operations  $\langle V_R(a) \rangle$  and  $[V_R(a)]$ . As  $V_R(a) \subseteq V(a)$ , we have the inclusions:  $\langle V_R(a) \rangle \subseteq \langle V(a) \rangle$  and  $[V_R(a)] \subseteq [V(a)]$ , for all  $a$ .

The relation “ $u \in \langle V(a) \rangle \Phi$ ” is read “some possible performance of  $V(a)$  at  $u$  yields a result in  $\Phi$ ” while “ $u \in \langle V_R(a) \rangle \Phi$ ” is read “some actual (or realizable) performance of  $V(a)$  at  $u$  yields a result in  $\Phi$ ”. These phrases bear different meanings.

In turn,  $u \in [V(a)]\Phi$  and  $[V_R(a)]\Phi$  are read “every possible performance of  $V(a)$  at  $u$  yields a result in  $\Phi$ ” and “every actual (or realizable) performance of  $V(a)$  at  $u$  yields a result in  $\Phi$ ”, respectively.

The above definitions are extended onto the words  $x \in \Sigma^*$  and the subsets  $A$  of  $\Sigma^*$ . Thus

$$(5) \quad u \in \langle V(x) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV(x)w \wedge w \in \Phi).$$

$$(6) \quad u \in [V(x)]\Phi \Leftrightarrow_{df} (\forall w)(uV(x)w \Rightarrow w \in \Phi).$$

$$(7) \quad u \in \langle V(A) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV(A)w \wedge w \in \Phi).$$

$$(8) \quad u \in [V(A)]\Phi \Leftrightarrow_{df} (\forall w)(uV(A)w \Rightarrow w \in \Phi).$$

Similarly,

$$(9) \quad u \in \langle V_R(x) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV_R(x)w \wedge w \in \Phi).$$

$$(10) \quad u \in [V_R(x)]\Phi \Leftrightarrow_{df} (\forall w)(uV_R(x)w \Rightarrow w \in \Phi).$$

$$(11) \quad u \in \langle V_R(A) \rangle \Phi \Leftrightarrow_{df} (\exists w)(uV_R(A)w \wedge w \in \Phi).$$

$$(12) \quad u \in [V_R(A)]\Phi \Leftrightarrow_{df} (\forall w)(uV_R(A)w \Rightarrow w \in \Phi).$$

It is easy to see that  $x = a_1 \dots a_m$ , then

$$(13) \quad \langle V(a_1 \dots a_m) \rangle = \langle V(a_1) \rangle \dots \langle V(a_m) \rangle.$$

$$(14) \quad [V(a_1 \dots a_m)] = [V(a_1)] \dots [V(a_m)],$$

where on the right-hand side the composition of the above unary operations occurs. Similar identities hold for  $\langle V_R(a_1 \dots a_m) \rangle$  and  $[V_R(a_1 \dots a_m)]$ .

It follows from Definition 3.6 that

$$(15) \quad \langle V_R(x) \rangle \Phi \subseteq \mathbf{Perf}(V_{seq}(x)), \text{ for all } \Phi \subseteq W.$$

Moreover

$$(16) \quad \mathbf{Perf}(V_{seq}(x)) = \langle V_R(x) \rangle W.$$

and

$$(17) \quad \mathbf{Perf} (V_{seq}(A)) = \langle V_R(A) \rangle W.$$

Thus the performability operators  $\mathbf{Perf} (V_{seq}(x))$  and  $\mathbf{Perf} (V_{seq}(A))$  are definable in terms of the above modal operators  $\langle V_R(x) \rangle$  and  $\langle V_R(A) \rangle$  and the constant representing truth, respectively.

The identities (16) and (17) show that (a version of) DL employing the above modal operators is capable of expressing the main definitions of performability of actions.

But we are concerned with Kleene’s closure of the relations  $V(x)$ ,  $x \in \Sigma^*$ . We therefore restrict compound actions only to those of the form  $V(A)$ , where  $A$  is a regular subset of  $\Sigma^*$ .

**Definition 9.1** Let  $\Sigma$  be an alphabet. The regular expressions over  $\Sigma$  and the languages they denote are defined recursively as follows (after Hopcroft and Ullman 1979).

If  $r$  is a regular expression,  $A(r) \subseteq \Sigma^*$  is the language  $r$  denotes. We assume:

1.  $\emptyset$  is a regular expression and it denotes the empty set,  $A(\emptyset) = \emptyset$ .
2.  $e$  is a regular expression and it denotes the set  $A(e) = \{e\}$ .
3. For every  $a \in \Sigma$ ,  $a$  is a regular expression denoting the set  $A(a) = \{a\}$ .
4. If  $r$  and  $s$  are regular expressions denoting the languages  $A(r)$  and  $A(s)$ , respectively, then  $(r + s)$ ,  $(rs)$ , and  $(r^*)$  are regular expressions that denote the sets  $A(r) \cup A(s)$ ,  $A(r)A(s)$ , and  $A(r)^*$ , respectively.

The regular expressions defined in 1.–3. are called *atomic*.

$\mathbf{Reg}(\Sigma)$  is the set of regular expressions over  $\Sigma$ . □

$A(r)$  is a set of words over  $\Sigma$ , i.e., it is a compound action, for all  $r$ . The discussion in this section is restricted to compound actions of the form  $A(r)$  for regular  $r$ .

$\mathbf{REG}(\Sigma)$  is the set of regular languages over  $\Sigma$ . Thus  $\mathbf{REG}(\Sigma) = \{A(r) : r \in \mathbf{Reg}(\Sigma)\}$ . Following the terminology adopted in this work, the set  $A(r)$  is called the *formal action* corresponding to  $r$ .

It is well known that  $\mathbf{REG}(\Sigma)$  forms a field of subsets of  $\Sigma^*$ . Thus it makes sense to introduce two secondary operations on regular sets, viz., the meet and the complement (to  $\Sigma^*$ ). One may accordingly enrich the above vocabulary of regular expressions by two more clauses:

5. If  $r$  and  $s$  are regular expressions denoting the languages  $A(r)$  and  $A(s)$ , respectively, then  $(r \cap s)$  and  $(r')$  are regular expressions that denote the sets  $A(r) \cap A(s)$ , and  $\Sigma^* - A(r)$ , respectively.

$\mathbf{Reg}(\Sigma)$  validates some equations holding for regular expressions.  $r = s$  means that  $A(r) = A(s)$ . Then it is easy to see that e.g.  $(p + q) = (q + p)$ ,  $((p + q) + r) = ((p + (q + r)))$ ,  $(p + q)r = (pr + qr)$ , etc. (Some parentheses are omitted.)

Models for formal regular actions are defined similarly as in Sect. 3.

**Definition 9.2** Let  $M = \langle W, V, V_R \rangle$  be an action model over  $\Sigma$ . Each non-empty word  $x = a_1 \dots a_m \in \Sigma^*$  is interpreted as the sequence of binary relations  $V_{seq}(x) := \langle V(a_1), \dots, V(a_m) \rangle$ , the *sequential action* of  $x = a_1 \dots a_m$ . The empty word  $e$  is

interpreted as the diagonal relation of  $W$ . In turn, every compound action  $A \subseteq \Sigma^*$  is interpreted as  $V_{seq}(A) = \{V_{seq}(x) : x \in A\}$ .

In the standard way (see Definitions 3.3 and 3.4), we define the binary relations  $V(x)$  and  $V(A)$ , being the resultant actions of  $x$  and  $A$  in the model, as well as the relations  $V_R(x)$  and  $V_R(A)$ , the realizable resultant relations of  $x$  and  $A$ .

In what follows, the focus is only on the compound actions of the form  $A(\mathbf{r})$ , where  $\mathbf{r}$  is a regular expression, and the sequential interpretation  $V_{seq}(A(\mathbf{r}))$  in the model. Other compound actions are disregarded. The set  $V_{seq}(A(\mathbf{r}))$  is defined by induction on complexity of regular expressions. Thus for every  $a \in \Sigma$ ,

$$\begin{aligned} V_{seq}(A(\mathbf{a})) &= V(a) (= \langle V(a) \rangle), \\ V_{seq}(A(\emptyset)) &= \emptyset, \\ V_{seq}(A(\mathbf{e})) &= V(e), \\ V_{seq}(A(\mathbf{r} + \mathbf{s})) &= V_{seq}(A(\mathbf{r})) \cup V_{seq}(A(\mathbf{s})), \\ V_{seq}(A(\mathbf{rs})) &= V_{seq}(A(\mathbf{r}))V_{seq}(A(\mathbf{s})) \quad (= \text{the set of all} \\ &\quad \text{concatenations of strings of relations of } V_{seq}(A(\mathbf{r})) \\ &\quad \text{and the strings of relations of } V_{seq}(A(\mathbf{s}))); \\ V_{seq}(A(\mathbf{r}^*)) &= (V_{seq}(A(\mathbf{r})))^* \quad (= \text{the Kleene closure of the set of all} \\ &\quad \text{sequences of relations belonging to } V_{seq}(A(\mathbf{r}))). \end{aligned}$$

The sets of the form  $V_{seq}(A(\mathbf{r}))$  are called *compound regular actions* in the model  $M$ .

In a similar manner, for every regular  $\mathbf{r}$  one defines the binary resultant relation  $V(A(\mathbf{r}))$  of  $V_{seq}(A(\mathbf{r}))$ . Thus

$$\begin{aligned} V(A(\mathbf{r} + \mathbf{s})) &:= V(A(\mathbf{r})) \cup V(A(\mathbf{s})), \\ V(A(\mathbf{rs})) &:= V(A(\mathbf{r})) \circ V(A(\mathbf{s})) \quad (= \text{the composition} \\ &\quad \text{of the relations } V(A(\mathbf{r})) \text{ and } V(A(\mathbf{s}))), \end{aligned}$$

and

$$V(A(\mathbf{r}^*)) := V(A(\mathbf{r}))^* \quad (= \text{the Kleene closure of the binary relation } V(A(\mathbf{r}))).$$

Moreover  $V(\mathbf{e})$  is the diagonal  $\mathbf{0}_W$  of  $W$  and  $V(\emptyset)$  is the empty set.

In turn,  $V_R$  is the mapping assigning to each atomic regular expression  $\mathbf{r}$  a sub-relation  $V_R(A(\mathbf{r}))$  of  $V(A(\mathbf{r}))$ , the realizable resultant relation of  $V_{seq}(A(\mathbf{r}))$ . The definition of  $V_R$  is inductively extended onto arbitrary regular expressions  $\mathbf{r} \in \mathbf{Reg}(\Sigma)$ .

Following Definition 3.4,  $V(A(\mathbf{r}))$  is called the *resultant relation* of  $V_{seq}(A(\mathbf{r}))$  on the set of states of  $W$ . Likewise,  $V_R(A(\mathbf{r}))$  is called the *realizable* (or *actual*) *resultant relation* of  $V_{seq}(A(\mathbf{r}))$  on  $W$ .

Given a model  $M = \langle W, V, V_R \rangle$  and a regular expression  $\mathbf{r}$  we want to define the proposition  $\mathbf{Perf} V_{seq}(A(\mathbf{r})) \subseteq W$ , for all  $\mathbf{r} \in \mathbf{Reg}(\Sigma)$ . If  $u \in W$ , then “ $u \in$

**Perf**  $V_{seq}(A(\mathbf{r}))$ ” is read:

“The action  $V_{seq}(A(\mathbf{r}))$  is performable at  $u$ .”

Let  $M = \langle W, V, V_R \rangle$  be a model and  $u \in W$  a state. Following the general pattern of performability introduced in Sect. 3, we accept the following two clauses:

- (1) Let  $\mathbf{r}$  be an atomic regular expression.  $V_{seq}(A(\mathbf{r}))$  is performable at  $u \Leftrightarrow$  there exists a state  $w$  such that  $uV_R(A(\mathbf{r}))w$  holds.
- (2)  $V_{seq}(A(\mathbf{r}+\mathbf{s}))$  is performable at  $u \Leftrightarrow V_{seq}(A(\mathbf{r}))$  is performable at  $u$  or  $V_{seq}(A(\mathbf{s}))$  is performable at  $u$ .
- (3)  $V_{seq}(A(\mathbf{r}^*))$  is performable at  $u \Leftrightarrow V_{seq}(A(\mathbf{r}^n))$  is performable at  $u$  for some  $n \geq 0$ .

Now, if we take the regular expression  $(\mathbf{rs})$ , then it is natural to postulate:

- (4)  $V_{seq}(A(\mathbf{rs}))$  is performable at  $u \Leftrightarrow V_{seq}(A(\mathbf{r}))$  is performable at  $u$  and moreover, after performing  $V_{seq}(A(\mathbf{r}))$  at  $u$ , a state  $w$  has been reached such that  $V_{seq}(A(\mathbf{s}))$  is performable at  $w$ .

(4) implies that for any action symbols  $a, b$ , the sequential action  $V_{seq}(A(\mathbf{ab}))$  is performable at  $u \Leftrightarrow$  the atomic action  $V_{seq}(A(\mathbf{a}))$  is performable at  $u$  and after performing  $V_{seq}(A(\mathbf{a}))$  at  $u$ , a state  $w$  has been reached such that  $V_{seq}(A(\mathbf{b}))$  is performable at  $w$ .

(4) offers a stronger notion of performability of sequential actions in comparison with Definition 3.5.(2). The reason is that we want to combine the notion of performability of regular actions with their algebraic structure encapsulated in Definition 9.1. (4) directly refers to the algebraic structure of regular actions. Moreover, the notion of performability should preserve the algebraic structure of  $\mathbf{Reg}(\Sigma)$  in the following sense: if  $\mathbf{r} = \mathbf{s}$ , then in any model  $M$  and any state  $u$ ,  $V_{seq}(A(\mathbf{r}))$  is performable at  $u$  if and only if  $V_{seq}(A(\mathbf{s}))$  is performable at  $u$ .

To achieve this, some care is needed when in action models  $M = \langle W, V, V_R \rangle$  one operates with the mapping  $V_R$  assigning to each symbol  $a \in \Sigma$  a subrelation  $V_R(A(\mathbf{a}))$  of  $V(A(\mathbf{a}))$ , the realizable resultant relation of  $V_{seq}(A(\mathbf{a}))$ . The same concerns the relations  $V_R(A(\mathbf{r}))$  defined for all  $\mathbf{r} \in \mathbf{Reg}(\Sigma)$ . We postulate that  $V_R(A(\mathbf{e})) = V(A(\mathbf{e})) (= \mathbf{0}_W)$ . In other words, the diagonal action  $V(A(\mathbf{e}))$  is performable in any state  $u \in W$ . This guarantees that  $V_R(A(\mathbf{re})) = V_R(A(\mathbf{e})) \circ V_R(A(\mathbf{r})) = V_R(A(\mathbf{r})) \circ V_R(A(\mathbf{e})) = V_R(A(\mathbf{er}))$ .

For any regular expression  $\mathbf{r}$ , the proposition  $\mathbf{Perf} V_{seq}(A(\mathbf{r})) \subseteq W$  is defined as follows:

$$u \in \mathbf{Perf} V_{seq}(A(\mathbf{r})) \Leftrightarrow V_{seq}(A(\mathbf{r})) \text{ is performable at } u.$$

The language  $L$  for action performability defined in Sect. 4 can be suitably accommodated to regular languages.  $L_{reg}$  is the language of performability of regular actions over  $\Sigma$ . Atomic formulas of  $L_{reg}$  are expressions of the form:

$$\mathbf{Perf}(\mathbf{r}), \mathbf{r} \in \mathbf{Reg}(\Sigma).$$

Compound formulas are formed from the atomic ones by applying the Boolean connectives  $\rightarrow$  and  $\neg$ . The other Boolean connectives are defined in the standard way. The language  $L_{reg}$  is poor. It contain neither propositional variables nor the modal connectives  $[r]$ ,  $\langle r \rangle$ ,  $r \in Reg(\Sigma)$ .

The above semantics validates some specific formulas for regular actions. Some of them are consequences of the above identities, as e.g.,

$$\begin{aligned} Perf(ep) &\leftrightarrow Perf(pe) \\ Perf(ep) &\leftrightarrow Perf(p) \\ Perf(p + q) &\leftrightarrow Perf(q + p) \\ Perf(p + q)r &\leftrightarrow Perf(pr + qr), \\ Perf(r^*) &\rightarrow Perf(r^n), \end{aligned}$$

for all  $n \geq 0$ , and many others. Specific action tautologies are, for example,

$$\begin{aligned} Perf(p + q) &\leftrightarrow Perf(p) \vee Perf(q), \\ Perf(pq) &\rightarrow Perf(p). \end{aligned}$$

But more importantly, the above semantics for regular actions validates the  $\omega$ -Rule:

$$\{Perf(r^n) : n \geq 0\} / Perf(r^*).$$

The  $\omega$ -Rule is generally not eliminable and cannot be replaced by a set of finitary rules. This fact makes the logic based on the above semantics infinitary. There is however a class of models for regular actions that make it possible to eliminate the  $\omega$ -Rule in the logic of action determined by this class.

$K_{reg}$  is the class consisting of models  $M = \langle W, V, V_R \rangle$  over  $\Sigma$  such that, for every  $a \in \Sigma$ ,  $V(a)$  is a unary total function from the set of states  $W$  to  $W$  and moreover  $V_R(a) = V(a)$ , for all  $a$ . Thus every atomic action  $V(a)$  is performable in all states  $u \in W$ . It is also postulated that  $V_R(e) = V(e) = \mathbf{0}_W$ . It then follows by induction on complexity of regular expressions  $r$  that every action  $V_{seq}(A(r))$ ,  $r \neq \emptyset$ , is performable at any state  $u \in W$ .

$\models_{reg}$  is the consequence relation in the language  $L_{reg}$  determined by the models of  $K_{reg}$ .

In the light of the above remark,  $\models_{reg}$  validates all formulas  $Perf(r)$ , for any non-zero regular expression  $r$ . This logic trivializes the problem of performability of regular actions.

The fact that every fomula  $Perf(A(r))$ ,  $r \neq \emptyset$ , is a tautology of  $\models_{reg}$  should not be confused with the problem of acceptability of regular languages by finite deterministic automata. The issue of acceptance of words by a FDA differs from the problem of its performability. If one treats such automata as action systems with the initial state and the final states discarded, one obtains models in the above sense over a finite  $\Sigma$ . In any finite deterministic automaton the atomic actions corresponding to the symbols of  $\Sigma$  are performable in all possible states. (This is not the case in non-deterministic automata.) Finite DAs are designed as devices in which transitions between states are

total functions labeled by the symbols of  $\Sigma$ . Thus acceptance of a word  $x$  and its performability are different problems!

Suitably modifying the proof of Theorem 7.5 one arrives at the following fact:

**Theorem 9.3** *The consequence relation  $\models_{reg}$  is finitary.* □

## 10 Final Remarks

### 10.1 Fuzzy Performability and ukasiewicz Logics

This work expounds the zero-one conception of performability—a given action is either performable in a state or it is not. A more refined and nuanced notion involving various grades of performability as e.g. in the system of school grades, would require introducing a modest amount of fuzzy sets theory (or—and this is yet another interesting option—many-valued logics) to action theory.

Let  $A$  be a complete lattice and  $W$  a non-empty set. Any mapping  $P : W \times W \rightarrow A$  is called an  $A$ -valued binary relation on  $W$ . If  $P$  and  $Q$  are  $A$ -valued relations on  $W$ , then  $P \circ Q$  is the  $A$ -valued relation on  $W$  defined as follows:

$$(P \circ Q)(u, w) := \sup\{\min(P(u, v), Q(v, w)) : v \in W\},$$

for all pairs  $\langle u, w \rangle \in W \times W$ .  $P \circ Q$  is called the *composition* of the relations  $P$  and  $Q$  with respect to  $A$ .

If  $A$  is the two-element lattice, then  $A$ -valued binary relations on  $W$  are ordinary binary relations on  $W$ . Moreover, the composition  $P \circ Q$  of  $A$ -valued relations coincides with the standard definition of composition.

To simplify matters, we chose the unit interval  $I$  as a fixed complete lattice we shall work with. By definition,  $I := \{r \in \mathbb{R} : 0 \leq r \leq 1\}$ .

One way of facing the problem of many-valuedness in action theory is in introducing real-valued models. According to this standpoint, a real-valued model (an  $I$ -based model, for short) is a triple  $\langle W, V, V_R \rangle$  consisting of the set  $W$  of states, a mapping  $V$  assigning to each symbol  $a \in \Sigma$  a function  $V(a) : W \times W \rightarrow I$ , and a mapping  $V_R$  assigning to each  $a \in \Sigma$  a function  $V_R(a) : W \times W \rightarrow I$  such that  $V_R(a) \leq V(a)$ , that is,

$$V_R(a)(u, w) \leq V(a)(u, w),$$

for all  $\langle u, w \rangle \in W \times W$ .

$V(a)$  and  $V_R(a)$  are thus  $I$ -valued relations on the set  $W$ .

Each function  $V(a)$ ,  $a \in \Sigma$ , is referred to as an  $I$ -valued atomic action on the set  $W$ .  $\{V(a) : a \in \Sigma\}$  is the family of  $I$ -valued atomic actions.

For each pair  $\langle u, w \rangle \in W \times W$ , the value  $V(a)(u, w) \in I$  measures the grade the pair  $\langle u, w \rangle$  is a possible performance of  $V(a)$ .

However, if one takes into account various forms of performability of actions, it is also necessary to introduce an  $I$ -valued transition relation between states. This relation is represented by the mapping  $V_R$ .

$V_R(a)(u, w)$  measures the *degree of performability* of  $V(a)$  for the input  $u$  and the output  $w$ . Thus if  $V_R(a)(u, w) = 1/3$ , then we shall say that  $1/3$  is the grade assigned to  $V(a)$  on passing from  $u$  to  $w$ . Given states  $u, w_1$  and  $w_2$ , if  $V_R(a)(u, w_1) < V_R(a)(u, w_2)$ , then we say that at the input  $u$ , the output  $w_2$  of  $V(a)$  receives higher degree of performability than the output  $w_1$  of  $V(a)$ .

Following the general pattern discussed in Sect. 3, every non-empty word  $x = a_1 \dots a_m$  is interpreted in the model as the following sequence of  $I$ -valued functions

$$V_{seq}(x) := \langle V(a_1), \dots, V(a_m) \rangle.$$

As to the interpretation of the empty word  $e$ , we first define the zero-one discriminator mapping  $\mathbf{0}_W$  on  $W \times W$ :

$$\mathbf{0}_W(u, w) := \begin{cases} 1 & \text{if } u = w \\ 0 & \text{otherwise.} \end{cases}$$

We then put:

$$V_{seq}(e) := \langle \mathbf{0}_W \rangle.$$

In what follows we shall identify  $V_{seq}(e)$  with  $\mathbf{0}_W$ . Every compound action  $A \subseteq \Sigma^*$  is interpreted as the set of sequences of functions:

$$V_{seq}(A) := \{V_{seq}(x) : x \in A\}.$$

The mapping  $V$  is inductively extended onto arbitrary words  $x \in \Sigma^*$  as follows:

$$V(e) := \mathbf{0}_W,$$

$V(xa)$  is the  $I$ -composition of the  $I$ -valued relations  $V(x)$  and  $V(a)$ :

$$V(xa) = V(x) \circ V(y),$$

that is,

$$V(xa)(u, w) := \sup\{\min(V(x)(u, v), V(a)(v, w)) : v \in W\},$$

for all pairs  $\langle u, w \rangle \in W \times W$ .

Note that then  $V(x) = V(xe) = V(ex)$ , for all words  $x \in \Sigma^*$ .

The above definitions imply that

$$V(xy)(u, w) := \sup\{\min(V(x)(u, v), V(y)(v, w)) : v \in W\},$$

for all words  $x, y$  and all pairs  $\langle u, w \rangle \in W \times W$ . This means that  $V(xy)$  is the  $I$ -composition of the  $I$ -valued sequential actions  $V(x)$  and  $V(y)$ , i.e.,  $V(xy) = V(x) \circ V(y)$ .

In the next step  $V$  is extended onto the power set of  $\Sigma^*$ . For the empty set  $\emptyset$ , it is declared that  $V(\emptyset)$  is the zero mapping, that is,

$$V(\emptyset)(u, w) := 0,$$

for all  $\langle u, w \rangle \in W \times W$ .

If  $A$  is a non-empty subset of  $\Sigma^*$ , then it is declared that

$$V(A)(u, w) := \sup\{V(x)(u, w) : x \in A\},$$

for all pairs  $\langle u, w \rangle \in W \times W$ .

In an analogous way  $V_R$  is extended onto words of  $\Sigma^*$  and the subsets of  $\Sigma^*$ . For the empty word  $e$ , it is assumed that

$$V_R(e)(u, w) \leq V(e)(u, w),$$

for all  $\langle u, w \rangle \in W \times W$ . Then, for any non-empty word  $x \in \Sigma^*$  and any  $a \in \Sigma$ :

$$V_R(xa) := V_R(x) \circ V_R(a),$$

that is,

$$V_R(xa)(u, w) := \sup\{\min(V_R(x)(u, v), V_R(a)(v, w)) : v \in W\},$$

for all  $\langle u, w \rangle \in W \times W$ .

We declare that

$$V_R(A)(u, w) := \sup\{V_R(x)(u, w) : x \in A\},$$

for all  $\langle u, w \rangle \in W \times W$ . (Thus  $V_R(A)$  is the supremum of the functions  $V_R(x), x \in A$ .)

Let  $M = \langle W, V, V_R \rangle$  be an  $I$ -based model. In the next step, to each  $I$ -valued action  $V(a)$  and each state  $u \in W$  an element  $\mathbf{Perf}_R(V(a), u) \in I$  is assigned which measures the degree of performability of  $V(a)$  ( $= V_{seq}(a)$ ) at  $u$  with respect to  $V_R$ . We may also say that  $\mathbf{Perf}_R(V(a), u)$  is the degree of  $V_R$ -performability of  $V(a)$  at  $u$  in the model  $M$ . We put:

$$\mathbf{Perf}_R(V(a), u) := \sup\{V_R(a)(u, w) : w \in W\},$$

for all  $a \in \Sigma$ . This definition is extended onto sequential actions  $V_{seq}(x)$  in the following way.

For the empty word  $e$ :

$$\mathbf{Perf}_R(V_{seq}(e), u) := \sup\{V_R(e)(u, w) : w \in W\},$$

$$\mathbf{Perf}_R(V_{seq}(x), u) := \sup\{V_R(x)(u, w) : w \in W\},$$

for any non-empty word  $x$ .

$\mathbf{Perf}_R(V_{seq}(x), u)$  measures the degree of performability of the sequence  $V_{seq}(x)$  at  $u$ .

If  $A \subseteq \Sigma^*$ , it is declared that:

$$\mathbf{Perf}_R(V_{seq}(A), u) := \sup\{\mathbf{Perf}_R(V_{seq}(x), u) : x \in A\}.$$

In particular,  $\mathbf{Perf}_R(V_{seq}(\emptyset), u) = 0$ .

To simplify the further discussion, we shall assume that the pertinent models  $M = \langle W, V, V_R \rangle$  are normalized, which means that  $V = V_R$ . The predicate  $\mathbf{Perf}_R$  is therefore simply marked as  $\mathbf{Perf}$ .

The above remarks does not preclude classical logic from the parlance about degrees of performability of actions. (By analogy, classical logic is not rejected when one speaks of school grades.) But then the language  $L$  of action performability should be essentially remodeled. According to this option, in the simplest case atomic formulas take the form of equations

$$\begin{aligned} \mathbf{Perf}(a) &\approx d, \\ \mathbf{Perf}(x) &\approx d, \\ \mathbf{Perf}(A) &\approx d, \end{aligned}$$

where apart from action variables  $a \in \Sigma$ ,  $x \in \Sigma^*$  and  $A$  ranging over subsets of  $\Sigma^*$ , we also have variables  $d$  (with indices, if necessary) ranging over the elements of  $I$ .

$\mathbf{Perf}(a) \approx d$  is read “The degree of performability of  $a$  is equal to  $d$ ”. Analogously one reads the other equations. They are classically evaluated: truth values (truth or falsity) of such formulas depend then on two parameters: particular states  $u$  of  $W$  and particular values assigned to the variable  $d$  in  $I$ . Accordingly,  $\mathbf{Perf}(a) \approx d$  is true at  $u$  for the degree  $r$  in an  $I$ -based model  $\langle W, V \rangle$  if and only if  $\mathbf{Perf}(V(a), u)$  is equal to the degree  $r$  assigned to the variable  $d$  in  $I$ . Analogously one defines the truth conditions for the remaining formulas. It is then clear how to extend the above truth-conditions on compound formulas within the framework of classical logic.

Yet another option is to employ many-valued logics. The best candidate is the infinitely-valued Łukasiewicz logic. According to this approach, the elements of  $I$  are treated as logical values. But this approach requires introducing the following additional assumption:

*For every  $I$ -based normalized model  $\langle W, V, V_R \rangle$  and any  $a \in \Sigma$ ,  $u \in W$ , the degree of performability  $\mathbf{Perf}(V(a), u)$  is identified with the logical value the atomic formula  $\mathbf{Perf}(a)$  receives at  $u$ .*

Analogously one interpretes the degrees  $\mathbf{Perf}(V_{seq}(x), u)$  and  $\mathbf{Perf}(V_{seq}(A), u)$ .

Here is a bunch of simple remarks on the problem.

The unit interval  $I = [0, 1]$  is endowed with the operations  $\rightarrow, \wedge, \vee, \otimes, \oplus, \neg$  defined as follows:

$$\begin{aligned} (\rightarrow)_L & a \rightarrow b := \min(1, 1 - a + b), \\ (\wedge)_L & a \wedge b := \min(a, b), \\ (\vee)_L & a \vee b := \max(a, b), \\ (\otimes)_L & a \otimes b := \max(0, a + b - 1), \\ (\oplus)_L & a \oplus b := \min(1, a + b), \\ (\neg)_L & \neg a := 1 - a, \end{aligned}$$

for all  $a, b \in I$ . They are called the *Lukasiewicz operations*. The operations  $\rightarrow, \wedge$  and  $\vee$  are successively called implication, conjunction and disjunction.  $\otimes$  is called the strong conjunction and  $\oplus$  is the weak disjunction.  $\neg$  is the negation operation.

The system  $\mathbf{A}_c := \langle I, \rightarrow, \wedge, \vee, \oplus, \otimes, \neg \rangle$  is called the infinite *Lukasiewicz algebra* (Wójcicki 1988). All the displayed operations are treated here as primitive operations of  $\mathbf{A}_c$  but they are definable in terms of the operations  $\rightarrow$  and  $\neg$  in the well-known manner. (One may also take  $\oplus$  and  $\neg$  as primitive operations, because  $a \rightarrow b = \neg a \oplus b$ ,  $a \vee b = \neg(\neg a \oplus b) \oplus b$ ,  $a \wedge b = \neg(a \vee \neg b)$ , and  $a \otimes b = \neg(a \oplus \neg b)$ , for all  $a, b \in I$ .) We shall therefore choose the operations  $\rightarrow$  and  $\neg$  as primitive and assume that the remaining operations are termwise definable. The algebra  $\langle I, \rightarrow, \neg \rangle$ , also denoted by  $\mathbf{A}_c$ , is therefore similar to the language  $\mathbf{L}$  of action performability defined in Sect. 4.

Let  $\langle W, V, V_R \rangle$  be an  $I$ -based normalized model. We extend the mapping  $V$  onto the set of formulas of  $\mathbf{L}$  by making use of the primitive operations  $\rightarrow, \neg$  of  $\mathbf{A}_c$ . The mapping  $V$  enables one to assign an element of  $I$  to each formula  $\sigma$  of  $\mathbf{L}$  and each state  $u \in W$ .

To this end we recursively define a mapping  $H_V : L \times W \rightarrow I$  called the *truth assignment*. If  $\sigma$  is a formula and  $u$  a state, then  $H_V(\sigma, u)$  is the logical value assigned to  $\sigma$  at the state  $u$ .  $H_V$  is defined as follows:

$$H_V(\text{Perf}(a), u) := \mathbf{Perf}(V(a), u),$$

for all  $a \in \Sigma$ .

$$H_V(\text{Perf}(x), u) := \mathbf{Perf}(V_{\text{seq}}(x), u),$$

for all words  $x \in \Sigma^*$ .

$$H_V(\text{Perf}(A), u) := \mathbf{Perf}(V_{\text{seq}}(A), u),$$

for all sets  $A \subseteq \Sigma^*$ . (Thus  $H_V(\text{Perf}(\emptyset), u) = 0$ .)

$H_V(\text{Perf}(a), u)$  defines the degree of truth of the formula  $\text{Perf}(a)$  at  $u$ . A similar remark applies to the remaining atomic formulas.

$H_V$  is recursively extend  $H_V$  onto compound formulas of  $L$  according to the definitions of the operations  $\rightarrow$  and  $\neg$  of the algebra of  $A_c$ . Thus

$$\begin{aligned}
 H_V(\phi \rightarrow \psi, u) &:= H_V(\phi, u) \rightarrow H_V(\psi, u), \\
 H_V(\neg\phi, u) &:= \neg H_V(\phi, u),
 \end{aligned}$$

for all formulas  $\phi$  and  $\psi$ . (The symbols of operations of the algebra  $A_c$  that occur on the RHS of the above equations are the same as the symbols of the corresponding connectives of  $L$ ; but this fact should not lead to confusion.)

The above many-valued semantics of action performability based on the unit interval  $I$  gives rise to a consequence relation  $\models_I$  operating on the set of all formulas of  $L$  in the following way. Let  $X$  be a set of formulas of  $L$  and  $\sigma$  a formula. We say that  $\sigma$  *logically follows* from  $X$ , in symbols:

$$X \models_I \sigma$$

if for every  $I$ -based model  $M = \langle W, V \rangle$  and every state  $u \in W$ ,

$$\inf\{H_V(\phi, u) : \phi \in X\} \leq H_V(\sigma, u).$$

(Here  $\leq$  marks the standard order in the interval  $I$ .)

The above definition is meaningful because the linear order  $(I, \leq)$  is a complete lattice.

It is easy to see that  $\models_I$  is a consequence relation on the set  $L$  of formulas. One can prove that  $\models_I$  is *not* finitary.

According to the above formula,  $\models_I$  is a logic that is defined semantically as a system preserving degrees of truth. Systems of this type are investigated in many-valued logic *sensu largo*.

The above many-valued option of action performability is left open for further scrutiny. □

**Acknowledgements** The author would like to thank for the support from the Beethoven 2016 German-Polish research Project “Permissions, Information and Institutional Dynamics, Obligations, and Rights”. Grant from the National Science Centre of Poland (BEETHOVEN, UMO-2014/15/G/HS1/04514).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

Czelakowski, J. (2019). Deontic logic, context free-grammars, and agency. To appear.

- Czelakowski, J. (1996). Elements of formal action theory. Essays on Logic and Artificial Intelligence. In Andre Fuhrmann & Hans Rott (Eds.), *Logic, action and information* (pp. 3–62). Walter de Gruyter.
- Czelakowski, J. (2001). *Protoalgebraic logics*. Trends in Logic (Vol. 10). Kluwer.
- Czelakowski, J. (2015). *Freedom and enforcement in action. Elements of formal action theory*. Trends in Logic (Vol. 42). Springer.
- Czelakowski, J. (2020). Deontology of compound actions. *Studia Logica*, 108(1), 5–47.
- Fikes, R. N., & Nilsson, J. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4), 189–208.
- Gelfond, M., & Lifschitz, V. (1998). Action languages. *Electronic Transactions on Artificial Intelligence*, 2, 193–210.
- Giunchiglia, E., & Lifschitz, V. (1998). An action language based on causal explanation: preliminary report. *Proc. AAAI-98/ Innovative Applications of Artificial Intelligence*, 623–630.
- Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., & Turner, H. (2004). Nonmonotonic causal theories. *Artificial Intelligence*, 153(1–2), 49–104.
- Hopcroft, J. E., & Ullman, J. D. (1979). *Introduction to automata theory, languages, and computation*. Addison-Wesley.
- Lee, J., Lifschitz, V., & Yang, F. (2013). Action Language BC: Preliminary Report. In *International joint conference on artificial intelligence (IJCAI)*, 983–989.
- Lifschitz, V., & Ren W. (2006). A modular action description language. In *Proceedings of national conference on artificial intelligence (AAAI-06)*, 853–859
- Nowakowska, M. (1979). *Teoria dzialania (Action Theory)*. In Polish. PWN, Warszawa
- Nowakowska, M. (1973). *Language of motivation and language of actions*. Mouton.
- Nowakowska, M. (1973a). Formal theory of actions. *Behavioral Science*, 18, 393–413.
- Parikh, R. (1978). The completeness of propositional dynamic logic. *Mathematical foundations of computer science 1978, (Lecture Notes in Computer Science 64)* (pp. 403–415). Springer Verlag.
- Seegerberg, K. (1980). Applying modal logic. *Studia Logica*, 39, 275–295.
- Seegerberg, K. (1982). The logic of deliberate action. *Journal of Philosophical Logic*, 11, 233–254.
- Wójcicki, R. (1988). *Theory of logical calculi*. Basic Theory of Consequence Operations. Kluwer.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.