Capabilities and Equivalence of Systems and a Preview of Related Algebraic Structure

Petr Pinkas, Jana Klečková

Department of Computer Science and Engineering, University of West Bohemia
pinkasp@kiv.zcu.cz, kleckova@kiv.zcu.cz

Abstract

In this paper, we propose a different approach to modeling of a system that focuses on the capability of the system to participate in interactions with its environment. Accordingly, we use a mathematical object to describe this capability. In addition, we show an equivalency (from an outer point of view) between capabilities and states. Furthermore, we more generally discuss the equivalence of systems where comparison of capabilities and bisimulation are related. Finally, we define a composite algebraic structure and then show that capabilities of the system are a solution of a system of equations in this algebraic structure.

1. Introduction

The effort to rigorously modeling critical systems and exploring their properties is becoming more and more common. This effort ranges from the UML [3] to the model checking technique [1]. The model checking technique is a mature formal method based on operational semantics and mainly on exhaustive analysis of state space by brute force. However, computer science still looks up to mathematics, where it is possible to work with entire spaces at once. Such possibility together with preserving a close connection to operational semantics would be ideal. One attempt in this direction was made in [5] where a π-calculus was introduced.

This paper focuses on CCS (Calculus of Communicating Systems) [4], the predecessor and basis of π-calculus. Moreover, we are interested only in a narrow subset of CCS that forms the fundament of modeling a system in a CCS approach.

The subset of CCS is described as follows. Let us have a set *Act* of actions and a set *K* of constants. Then, a set *3* of expressions is recursively defined as follows:

$$3 = \{\ A, \alpha{\cdot}E, \sum_{i \in I} E_i\ :\ A \in K; \alpha \in Act; E, E_i \in 3\ \}$$

where operations $\cdot$ and $+$ are implicitly introduced. It is obvious that a set *3* is closed under both operations by definition. Consequently, the tuple $(3, +, \cdot)$ forms a simple algebraic structure and we can write e.g.

$$E = \alpha{\cdot}(F + \beta{\cdot}A) \qquad E, F \in 3 \quad \alpha, \beta \in Act \quad A \in K$$

So far, expressions from *3* have no meaning. Hence, the meaning is given by the transition system (*3*, *Act*, *Я*), where *Я* is the transition relation (i.e. a subset of $3 \times Act \times 3$) generated by two transition rules. These rules (related to operations $\cdot$ and $+$, respectively) are defined as follows:

$$\alpha \in Act\ \wedge\ E \in 3 \quad \Leftrightarrow \quad (\alpha{\cdot}E, \alpha, E) \in Я$$

$$j \in I\ \wedge\ (E_j, \alpha, F) \in Я \quad \Leftrightarrow \quad (\textstyle\sum_{i \in I} E_i, \alpha, F) \in Я$$

that means, for example, an expression $\alpha{\cdot}E$ becomes E by executing an action α (i.e. by participating on a proper handshake). See details in [4]. Despite the previous effort, constants from *K* still have no meaning. Moreover, elements from *3* remain as syntax formations (with no other mutual relations than the two rules defined above). For illustration, let us have a system given by the transition graph

and let us try to express it by our (so far described) subset of CCS. Thus, we make two attempts:

1) $A = \alpha{\cdot}A$ 2) $E = \alpha{\cdot}E$ $\qquad \alpha \in Act \quad A \in K \quad E \in 3$

which result into a failure. Both equations are invalid in terms of the mentioned algebraic structure (e.g. *A* and $\alpha{\cdot}A$ are different elements from *3*). Even the mentioned rules cannot help. First, we know that $\alpha{\cdot}A$ can become *A*, but we do not know if $\alpha{\cdot}A$ equals *A* (and consequently if $\alpha{\cdot}A$ can become $\alpha{\cdot}A$) because

we have no meaning of A. Second, it is clear that $\alpha{\cdot}E$ cannot become $\alpha{\cdot}E$ for any E from $3 \setminus K$. In [4] it is solved by a workaround, i.e. by a defining equation

$$A \overset{\text{def}}{=} \alpha{\cdot}A$$

that gives a meaning to the constant A. In addition, normal equations (e.g. $E + F = F + E$) are meant as comparison of derivatives (i.e. transition trees, see [4]). Now, with defining and derivatives equations, our subset of CCS is complete.

However, it is obvious that the CCS is no longer an algebraic structure.

2. Transition system

In the CCS approach, the definition of transition relation follows the structure of expressions and thus gives the semantics to them [4]. Nevertheless, in our approach, a transition relation and its meaning is the basis. Everything else is obtained by exploration and formalization of properties of this basis.

At first, however, we formally define a transition system as a tuple

$$(S, \Lambda, R^\Lambda)$$

where S is a finite set of states, Λ is a finite set of actions and R^Λ is a transition relation. More accurately, $R^\Lambda \subseteq S \times \Lambda \times S$, whereas

$$(A, \alpha, B) \in R^\Lambda \qquad A, B \in S \qquad \alpha \in \Lambda$$

if and only if exists a transition from state A to state B labeled with α. The transition system is always just in one of its states, i.e. it may arise at most one transition at once. Thus, execution has the form of a sequence of transitions.

Now we give a meaning to our transition system. Each transition represents an inner (causal) act of a modeled system. A part of the inner act is always exposed to the environment of the modeled system and this part is represented by the action in our transition system. However, the action is not meant as an outer symptom of the inner act; the action is meant as a complement in an interaction (i.e. it can be seen as the one side in a handshake). Therefore, the action can be executed only together with its complementary action from the environment (usually from the other system). Consequently, the execution of the action is a necessary condition for execution of a rest of the inner act.

3. Modeling and capabilities

In this section, we improve the semantics of our (inner) transition system. In addition, we develop an appropriate outer transition system (i.e. a description of a modeled system from an outer point of view). Finally, we introduce the capabilities (i.e. an outer transition system expressed purely by the actions).

3.1. Permissible executions

Usually, there are possible many executions (i.e. sequences of transitions) of our transition system but only some of them well characterize what a modeled system can do. Hence, we pay an attention to what a transition system permits, instead of what is possible (as an execution). It lead us (at first intuitively) to introduce a permissible executions. For example, let us have two transition systems
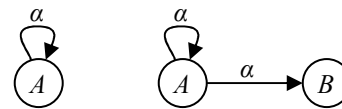
in states A. Both transition systems have one infinite possible execution (with sequence of actions $\alpha\alpha\alpha\dots$) and infinitely many finite possible executions (with sequences of actions α, $\alpha\alpha$, $\alpha\alpha\alpha$, $\dots$). Does it mean they are equal? No, the transition system on the left permits just the mentioned infinite execution, which means the modeled system is still capable to execute the action α. However, the transition system on the right permits only the mentioned finite executions. It means the modeled system is capable to execute any number of actions α, but not infinitely many (note that the number of actions is not predictable).

Accordingly, we formally define a permissible execution as a particular (and possibly infinite) walk on transition graph, i.e. as a sequence of transitions

$$(\ (X_i, \alpha_i, X_{i+1})\ )_{i \in I} \qquad X_1 = A \qquad (X_i, \alpha_i, X_{i+1}) \in R^\Lambda$$

where R^Λ is a transition relation, A is a non-terminal initial state of execution (note that a terminal state is a state which has no outgoing transitions) and I is a set of indices that is

- finite, i.e. $I = \{\ 1, 2, \dots, n\ \}$, if and only if a state X_{n+1} is a terminal state, and

- infinite, i.e. $I = \{\ 1, 2, \dots\ \}$, if and only if there exists at least one index i such that any terminal state is not reachable from state X_i.

For a given transition relation R^A and a given initial state A, we denote a set of all permissible executions as $\mathcal{C}^A_A$ (note that $\mathcal{C}^A_A$ is empty for an initial state A which is also a terminal state, i.e. there is no empty execution).

3.2. Interpretations of actions

The above-introduced set $\mathcal{C}^A_A$ (of all permissible executions) describes what a modeled system in a state A can do. However, we need a corresponding description from an outer (i.e. from the environment of a modeled system) point of view. In other words, we would like to abstract from any inner acts.

Hence, we attempt to obtain such description by defining a set Π^A_A of all permissible sequences of actions as follows:

$$\Pi^A_A = \{ \ (\alpha_i)_{i \in I} \ : \ (\ (X_i, \alpha_i, X_{i+1}) \ )_{i \in I} \in \mathcal{C}^A_A \ \}$$

i.e. Π^A_A represents sequences of actions which results from all executions that a transition system (in a state A) permits. However, this description has an issue which results from the nondeterminism of transition systems. For example, the two transition systems
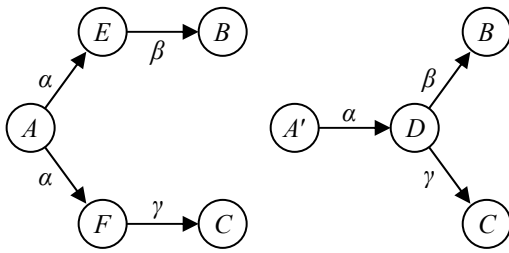
have the same set of all permissible sequences of actions, i.e.

$$\Pi^A_A = \Pi^A_{A'} = \{ \ \alpha\beta, \ \alpha\gamma \ \}$$

which is not correct. Indeed, the transition system on the right always permits to execute the sequence $\alpha\beta$ (and analogically the sequence $\alpha\gamma$) but the transition system on the left need not permit to execute it. More accurately, the execution of action α can induce the transition to the state F where the transition system does not permit to execute the action β. However, the sets Π^A_A and $\Pi^A_{A'}$ do not show such difference.

So, let us explore what happened from a modeled system point of view. The modeled system (in some state abstracted as A) is capable to participate on an interaction, via the action α, with its environment. More accurately, the modeled system can execute some two inner acts (abstracted as transitions to E and F) whose exposed parts are equal, i.e. both acts

expose the same action α to the environment. Thus, the two inner acts are indistinguishable in terms of interaction. It depends on the modeled system which of inner acts is induced by an upcoming interaction (note that we do not attempt to model it by our transition system, i.e. we leave it undetermined). However, the two inner acts do different things and mainly, they have different consequences. We cannot fully abstract from them (as we can in automata). From the outer point of view, there may be different consequences when the action occurred. In other words, the action is somehow interpreted by the modeled system and we should still reflect this. Hence, we introduce an interpretation of an action (shortly an interpretation) as a tuple (α, Φ) written for clarity as

$$\alpha^\Phi \qquad \alpha \in \Lambda \qquad \Phi \subseteq \Lambda$$

whereas an interpretation α^Φ represents an action α together with its consequence. In addition, we define a set Ψ of interpretations as follows:

$$\Psi = \{ \ \alpha^\Phi \ : \ \alpha \in \Lambda, \ \Phi \subseteq \Lambda \ \}$$

Now, as a formal foundation for next definitions, we define an extended transition system as a tuple

$$(S, \Psi, R^\Psi)$$

where S is an identical set of states as in original transition system (S, Λ, R^A), Ψ is a finite set of interpretations and R^Ψ is a transition relation. More accurately, $R^\Psi \subseteq S \times \Psi \times S$, whereas

$$(A, \alpha^{\Phi(B)}, B) \in R^\Psi \quad \Leftrightarrow \quad (A, \alpha, B) \in R^A$$

and $\Phi(X)$ is a set of actions of all outgoing transitions from a state X, which is defined as follows:

$$\Phi(X) = \{ \ \beta \ : \ (X, \beta, Y) \in R^A \ \}$$

It is obvious that an extended transition system still fully represents an original one, i.e.

$$(S, \Psi, R^\Psi) \quad \Leftrightarrow \quad (S, \Lambda, R^A)$$

because we only add some redundant (but useful) information. Not surprisingly, an extended transition system also preserves a meaning.

3.3. Postponed nondeterminism

Despite the effort in the previous subsection, we still cannot obtain a description from the outer point of view because the extended transition system still is not deterministic (in terms of interpretations) in general. Let us explore an example
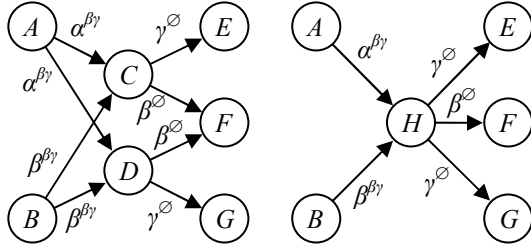
where the extended transition system on the left is nondeterministic for the interpretations $\alpha^{\beta\gamma}$ and $\beta^{\beta\gamma}$ in the states A and B respectively (note that we write all interpretations, e.g. $\alpha^{\{\beta,\gamma\}}$, for clarity in the form $\alpha^{\beta\gamma}$). However, the states C and D are not distinguishable from the outer point of view. Indeed, these states are reachable by the same actions (α, β) from the same states (A, B) and importantly, there are permitted the same following actions (β, γ) in these states. Hence, we can merge them, and thus we get a transformed transition system as shown on the right. Note that, in fact, we postpone the nondeterminism from the states A and B to the merged state H.

The previous example describes situations where for each examined state Y_i (e.g. C and D) a following condition is satisfied: if there exists some transition (X, α^Φ, Y_i), then there exists (X, α^Φ, Y_j) for any $j \neq i$. A following example describes situations where this condition is not satisfied. So, let us have an extended transition system
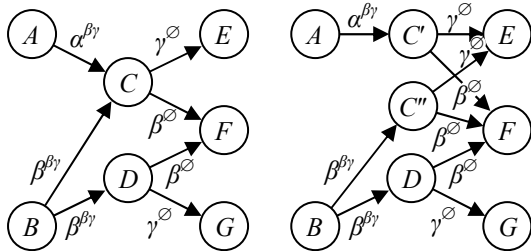
on the left. It is obvious that the state C does not satisfy the mentioned condition. Hence, we split it into the states C' and C'', and thus we again obtain a transformed transition system as shown on the right. After that, we can merge the states C'' and D (as in the previous example), and thus we again postpone the nondeterminism from the state B.

It should be noted that the initial state (if some is defined) need a special care. The initial state, e.g. A, has to be worked as if there exists an extra transition $(\partial, \partial^{\Phi(A)}, A)$ where ∂ and ∂ is not used by other transitions. With this assumption, the split and merge transformations (introduced above) can be applied even on the initial state.

We postpone the nondeterminism as long as it is not eliminated. It can be shown that we can eliminate the nondeterminism from any extended transition system. We denote the resulting transition system as an outer transition system that is defined as a tuple

$$(Q, \Psi, L^\Psi)$$

where Q is a finite set of states (i.e. a set S after all split and merge transformations), Ψ is a finite set of interpretations and L^Ψ is a transition relation (i.e. a relation R^Ψ after all split and merge transformations). A relationship between the extended transition system and the appropriate outer transition system is just an implication, i.e.

$$(S, \Psi, R^\Psi) \Rightarrow (Q, \Psi, L^\Psi)$$

because we (in general) loose informations of inner acts. However, the appropriate extended and outer transition systems are not distinguishable from the outer (i.e. from the environment of a modeled system) point of view. Moreover, an outer transition system is deterministic (in terms of interpretations).

3.4. Capability and state

As an outer transition system is deterministic, it allows us to obtain a description of what a modeled system in a given state can do from the outer point of view.

First, for a given transition relation L^Ψ and a given initial state A, we (analogically to $\mathcal{C}^A_A$) denote a set of all permissible executions as $\mathcal{C}^\Psi_A$.

Now, for a given transition system (Q, Ψ, L^Ψ) and a given state A from a set Q, we define a set Π^Ψ_A of all permissible sequences of interpretations as follows:

$$\Pi^\Psi_A = \{ (\alpha_i^{\Phi(Xi+1)})_{i \in I} : ((X_i, \alpha_i^{\Phi(Xi+1)}, X_{i+1}))_{i \in I} \in \mathcal{C}^\Psi_A \}$$

i.e. Π^Ψ_A represents sequences of interpretations which results from all executions that an outer transition system (in a state A) permits. This set is the needed description, and we denote it as a capability of a modeled system in a particular state.

Further, we explore a relationship between states and capabilities. It is obvious that

$$A \Rightarrow \Pi^{\psi}_{A}$$

which means that a state A determines a capability Π^{ψ}_{A} within a particular outer transition system. In addition, we denote states as equal if and only if they determine a same capability. For example, an outer transition system
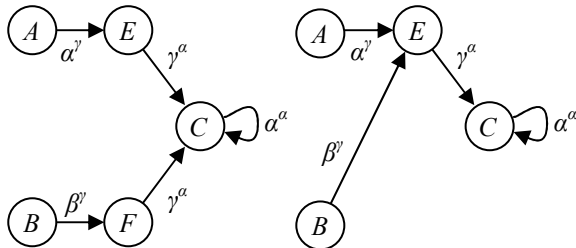
on the left has two equal states (i.e. the state E and F) because appropriate capabilities are

$$\Pi^{\psi}_{E} = \Pi^{\psi}_{F} = \{\ \gamma^{\alpha}\alpha^{\alpha}\alpha^{\alpha}\alpha^{\alpha}\dots\ \}$$

that means the transition system can execute just and only the same actions with the same consequences in both cases. Then it does not matter in which of these states the transition system is. Thus, we can merge these states without impact on any capability (note that it is similar procedure as in automata) and we get a reduced outer transition system as shown on the right. Consequently, after merging of all equal states, there does not exist any two different states X and Y that determine the same capability, i.e.

$$A \Leftrightarrow \Pi^{\psi}_{A}$$

Thus, a capability both represents an outer state of a modeled system and describes what a modeled system in such state can do from the outer point of view.

It becomes clear that we need to work with infinite sequences. In this paper, we use the notion Ψ^{∞} to denote the (infinite) set of all sequences of interpretations, i.e.

$$\Psi^{\infty} = \Psi^{*} \cup \Psi^{\omega}$$

where Ψ^{*} (Kleene closure) is the set of all finite sequences and Ψ^{ω} is the set of all infinite sequences over the set Ψ.
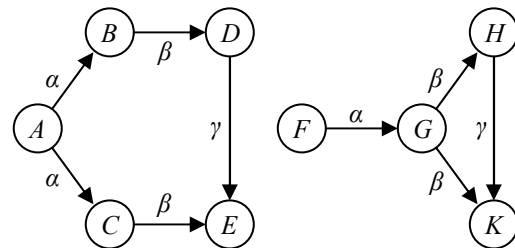
4. Equivalence

In the previous section, the notion of capability has been developed to be a full characterization of a modeled system (in a particular state) from an outer (i.e. from the environment of a modeled system) point of view. Hence, capabilities should be useful in determining the equivalence of systems.

In this section, we will more generally discuss the equivalence of such systems that can be modeled by a transition system defined in the section 2 (note that such systems can also be described by capabilities). However, when are such systems equal? Systems are regarded as equal if they are not distinguishable from each other from an outer point of view. Precisely, systems are not equal if and only if there is some experiment that distinguishes them [2]. Of course, the main role plays an experiment scenario. Since the examined system can interact with its environment only by actions from a set Λ, some scenarios based on "button-pushing" are appropriate.

In the further discussion about the equivalence, we will use a convenient example with two systems given by the transition graphs

4.1. Experimenting

In [2], there are more details about experimenting with equivalence. However, we shortly describe only the button-pushing experiments.

In these experiments, a system is seen as a black box with buttons labeled with the actions (from a set Λ), and no other controls. If the system can execute the action α, then it is possible to press the α-button and then the system will change state. If the system cannot execute α, then the α-button is locked; but the experimenter can press the button and discover that the system cannot execute the action α (and continue experimenting).

For the first experimenting, we use a lighted-button scenario. The buttons have lights inside them and the light on the button is lit when that button is locked. In other words, the experimenter can see at every stage, which actions are possible and which are not, without changing the state of the system. In addition, the experimenter can reset the system (i.e. restore the state that the system had before the experimenting) at any stage. Formally, a lighted-button experiment is a sequence $L_0 \alpha_0 L_1 \dots \alpha_n L_{n+1}$

alternating between sets L_i of locked buttons and α_i-buttons where $\alpha_i \notin L_i$. Hence, the sequences

$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\}$$
$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\} \;\; \gamma \;\; \{\,\alpha,\beta,\gamma\,\}$$
$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta,\gamma\,\}$$

are all experiments that can be done on both systems from our example. In other words, there is no one in the experiments, which can distinguish both systems. Hence, both systems are equal in terms of lighted-button experiment.

For the next experimenting, we use a duplicator scenario. It is a modified lighted-button scenario in which the experimenter can (instead of resetting the system) make any number of copies of the system at any stage, and perform independent experiments on the copies. Note that using of copies is the same thing as if the experimenter could restore the state that the system had in any previous stage (i.e. not only before the experimenting at all). Formally, the trees

$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\} \;\; \raisebox{0pt}{$\gg$}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\dots$$

$$\{\,\beta,\gamma\,\} \;\; \alpha \;\; \{\,\alpha,\gamma\,\} \;\; \raisebox{0pt}{$\gg$}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta,\gamma\,\}$$
$$\raisebox{0pt}{$\gg$} \;\; \{\,\alpha,\gamma\,\} \;\; \beta \;\; \{\,\alpha,\beta\,\}$$
$$\dots$$

are two experiments that could be done on the left-hand and right-hand system (respectively) from our example (note that $\gg$ represents duplication of the system). Indeed, the copies of the original systems made after executing α will show differences. The left-hand copies never prevent the experimenter from executing γ after β (because it seems the original system is in the state B). However, the right-hand copies prevent him but only sometimes (because the original system is surely in the state G). Hence, the systems certainly are not equal in terms of duplicator experiment.

4.2. Using bisimulation

Bisimulation equivalence is intended to identify transition systems with the same branching structure, and which thus can simulate each other in a stepwise manner [1]. As there are various formal definitions of bisimulation in [1, 4, 7], we appropriately rephrase the definition to the following form.

For given transition systems $(S_1, \Lambda, R^{\Lambda}_1)$ and $(S_2, \Lambda, R^{\Lambda}_2)$, a bisimulation is a relation $M \subseteq S_1 \times S_2$ with the following property:

$\forall\,(X, Y) \in M$:

(a) $\forall\,(X, \alpha, X') \in R^{\Lambda}_1 \quad \exists\,(Y, \alpha, Y') \in R^{\Lambda}_2 \;$:
$$(X', Y') \in M$$

(b) $\forall\,(Y, \alpha, Y') \in R^{\Lambda}_2 \quad \exists\,(X, \alpha, X') \in R^{\Lambda}_1 \;$:
$$(X', Y') \in M$$

Consequently, the transition systems in the states $A_1 \in S_1$ and $A_2 \in S_2$ are bisimulation equivalent (or shortly bisimilar), denoted $A_1 \sim A_2$, if there exists a bisimulation M such that $(A_1, A_2) \in M$.

Now, we apply bisimulation equivalence to our example. It is necessary to find a bisimulation M and thus let us try the following relation

$$\{\;(A, F),\; \textbf{(B, G)},\; \textbf{(C, G)},\; (D, H),\; (E, K)\;\}$$

However, the tuples (B, G) and (C, G) do not satisfy the condition (b) from the definition. It is because there are missing the tuples (D, K) and (E, H) in the relation. We add them. Thus, let us try the following relation

$$\{\;(A, F),\; (B, G),\; (C, G),\; (D, H),\; (E, K),$$
$$\textbf{(D, K)},\; \textbf{(E, H)}\;\}$$

Nevertheless, the new tuples (D, K) and (E, H) do not satisfy the conditions (a) and (b), respectively. This is a conflict; a bisimulation containing the tuple (A, F) cannot be found. Hence, the systems definitely are not equal in terms of bisimulation (or shortly, they are not bisimilar).

Consequently, bisimulation does not correspond to the lighted-button experiment in general. In [4], however, is stated: "…we only wish to distinguish between two agents P and Q if the distinction can be detected by an external agent interacting with each of them…" and a lighted-button experiment is used to advocate the needs of bisimulation. Similarly, in [7] is stated: "This is the notion of strong bisimulation, which takes the view that the only thing we can detect about a given process state is what events it can do, and that to be equivalent two processes must have the same set of events available immediately, with these events leading to processes that are themselves equivalent." Nevertheless, our example shows that bisimulation is too strong for the cited purposes (i.e. for lighted-button scenario).

4.3. Using capabilities

The notion of capability has been developed to be a full characterization of a modeled system (in a particular state) from an outer point of view. Thus, it should be enough to compare the particular

capabilities for determining the equivalence. Let us apply this idea to our example.

First, we should derive the sets $\mathcal{C}^{\Psi}_A$ and $\mathcal{C}^{\Psi}_F$ of all permissible executions for given transition systems in the states A and F, respectively. These sets are as follows:

$$\mathcal{C}^{\Psi}_A = \{ \ (A, \alpha^{\beta}, B) \ (B, \beta^{\gamma}, D) \ (D, \gamma^{\varnothing}, E),$$
$$(A, \alpha^{\beta}, C) \ (C, \beta^{\varnothing}, E) \ \}$$

$$\mathcal{C}^{\Psi}_F = \{ \ (F, \alpha^{\beta}, G) \ (G, \beta^{\gamma}, H) \ (H, \gamma^{\varnothing}, K),$$
$$(F, \alpha^{\beta}, G) \ (G, \beta^{\varnothing}, K) \ \}$$

Now, we can derive the capabilities Π^{Ψ}_A and Π^{Ψ}_F (i.e. sets of permissible sequences of interpretations) from the $\mathcal{C}^{\Psi}_A$ and $\mathcal{C}^{\Psi}_F$, respectively. The capabilities are as follows:

$$\Pi^{\Psi}_A = \{ \ \alpha^{\beta}\beta^{\gamma}\gamma^{\varnothing}, \ \alpha^{\beta}\beta^{\varnothing} \ \} \qquad \Pi^{\Psi}_F = \{ \ \alpha^{\beta}\beta^{\gamma}\gamma^{\varnothing}, \ \alpha^{\beta}\beta^{\varnothing} \ \}$$

Hence, the systems are equal in terms of capability. Unlike the bisimulation, the comparing of particular capabilities matches to the lighted-button experiment here. Of course, it is not a proof that it correspond in general. We are going to proof it in the further work.

5. Algebraic structure

In this section, we define a composite algebraic structure based on a set of capabilities (i.e. not of the syntax expressions as the calculus in CCS). Thus, a set K of capabilities is defined as follows:

$$K = 2^{\Lambda} \qquad \Lambda = \Psi^{\infty}$$

that means K is a (infinite) set of all subsets of a set Ψ^{∞} of all sequences of interpretations. Note that a capability Π^{Ψ}_X is a subset of a set Ψ^{∞}, i.e. we can write

$$\Pi^{\Psi}_X \in K$$

Note that K is determined only by a given set Λ of actions. That means not all capabilities from K are capabilities of a particular outer transition system. Simply, a set K is a universe of capabilities (for given sets Λ).

Further, we introduce variables that take values from K. These variables are written as letters in bold (i.e. A, B, C, …) and their semantics is defined as follows:

$$A = \Pi^{\Psi}_A$$

that means the variable A represents both an outer state A and a capability of an outer transition system in this state.

Now, we define the structure $(K, +)$ as an abelian monoid. Thus, the operation $+$ has to be a function $K \times K \rightarrow K$ that satisfies axioms of closure, identity, associativity and commutativity. Accordingly, we define $+$ as the union of two capabilities, i.e. we can write an equation $C = A + B$ as

$$\Pi^{\Psi}_C = \Pi^{\Psi}_A \cup \Pi^{\Psi}_B$$

and it is easy to show that our operation $+$ satisfies mentioned axioms:

- Closure axiom – Both Π^{Ψ}_A and Π^{Ψ}_B are subsets of Ψ^{∞}. Further, K is a power set of Ψ^{∞}, i.e. K contains all subsets of Ψ^{∞}. Finally, because the union of subsets is still a subset, K contains also Π^{Ψ}_C. Thus, a capability $A + B$ is from K for any capabilities A and B.

- Identity axiom – A set K (as a power set) contains the empty set, which means no capability. So, we introduce the identity element 0 (written as zero in bold) defined as follows:

$$0 = \varnothing$$

Thus, because the union of a set and the empty set is a set itself, the equation $A + 0 = 0 + A = A$ is valid for any capability A.

- Associativity and commutativity axiom – As the union operation is associative and commutative, the equations $(A + B) + C = A + (B + C)$ and $A + B = B + A$ are valid for any capabilities A, B and C.

Thus, we have obtained a simple algebraic structure but we need one more operator.

Therefore, we define the composite algebraic structure $(K, +, \Lambda, \cdot)$ as an extension of the structure $(K, +)$. The operation $\cdot$ is a function $\Lambda \times K \rightarrow K$ that satisfies the axiom of closure and right-associativity. Accordingly, we define $\cdot$ as the concatenation of an interpretation and a capability, i.e. we can write an equation $B = \alpha \cdot A$ as

$$\Pi^{\Psi}_B = \{ \ \alpha^{\Phi(A)}\zeta : \zeta \in \Pi^{\Psi}_A \ \} \qquad \Pi^{\Psi}_A \neq \varnothing$$

$$\Phi(A) = \{ \ \alpha : \alpha^{\Phi}\xi \in \Pi^{\Psi}_A \ \}$$

or

$$\Pi^{\Psi}_B = \{ \ \alpha^{\varnothing} \ \} \qquad \Pi^{\Psi}_A = \varnothing$$

If we use some properties of regular and ω-regular languages discussed in [1], it is again easy to show that our operation · satisfies the closure axiom. First, if ζ is from Ψ^*, then $\alpha^{\Phi(A)}\zeta$ is from Ψ^*, because the concatenation of two regular languages is another regular language. Second, if ζ is from Ψ^ω, then $\alpha^{\Phi(A)}\zeta$ is from Ψ^ω, because the concatenation of a regular and an ω-regular language is another ω-regular language. In all cases, both $\alpha^\varnothing$ and $\alpha^{\Phi(A)}\zeta$ are from Ψ^∞, i.e. Π^Ψ_B is a subset of Ψ^∞, and then Π^Ψ_B is from K. Thus, a capability $\alpha \cdot A$ is from K for any action α and any capability A.

Now, for illustration, we present our algebraic structure in a few interesting examples. First, let us have a system given by the transition graph

that we express by the simple equation

$$A = \alpha \cdot A$$

whose solution (the capability) is the following set of one permissible sequence of interpretations:

$$A = \{ \alpha^\alpha \alpha^\alpha \alpha^\alpha \ldots \}$$

Subsequently, we check this solution by substituting it into the right side of the equation, i.e.

$$\alpha \cdot A = \{ \alpha^\alpha \alpha^\alpha \alpha^\alpha \ldots \}$$

and we obtain the same set as A. Thus, the solution is correct. Indeed, if the infinite sequence $\alpha^\alpha \alpha^\alpha \alpha^\alpha \ldots$ is prefixed by one interpretation α^α, it is still just the infinite sequence $\alpha^\alpha \alpha^\alpha \alpha^\alpha \ldots$. Second, let us have a little different system, i.e. the transition graph, the related equation, and its significantly different solution as follows:
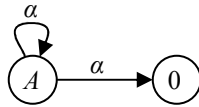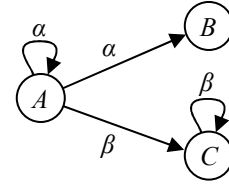

$$A = \alpha \cdot A + \alpha \cdot 0$$

$$A = \{ \alpha^\varnothing, \alpha^\alpha \alpha^\varnothing, \alpha^\alpha \alpha^\alpha \alpha^\varnothing, \ldots \}$$

Subsequently, we check the solution by substitution:

$$\alpha \cdot A = \{ \alpha^\alpha \alpha^\varnothing, \alpha^\alpha \alpha^\alpha \alpha^\varnothing, \ldots \} \qquad \alpha \cdot 0 = \{ \alpha^\varnothing \}$$

$$\alpha \cdot A + \alpha \cdot 0 = \{ \alpha^\varnothing, \alpha^\alpha \alpha^\varnothing, \alpha^\alpha \alpha^\alpha \alpha^\varnothing, \ldots \}$$

At last, let us have a more complex system, i.e. the transition graph, the related system of equations, and its solution (three capabilities) as follows:

$$A = \alpha \cdot A + \alpha \cdot B + \beta \cdot C$$
$$B = 0$$
$$C = \beta \cdot C$$

$$A = \{ \alpha^\varnothing, \alpha^{\alpha\beta}\alpha^\varnothing, \ldots, \beta^\beta \beta^\beta \ldots, \alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots,$$
$$\alpha^{\alpha\beta}\alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots, \ldots \}$$

$$B = \varnothing \qquad C = \{ \beta^\beta \beta^\beta \ldots \}$$

Subsequently, we check the solution by substitution:

$$\alpha \cdot A = \{ \alpha^{\alpha\beta}\alpha^\varnothing, \alpha^{\alpha\beta}\alpha^{\alpha\beta}\alpha^\varnothing, \ldots, \alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots,$$
$$\alpha^{\alpha\beta}\alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots, \ldots \}$$

$$\alpha \cdot B = \{ \alpha^\varnothing \} \qquad \beta \cdot C = \{ \beta^\beta \beta^\beta \ldots \}$$

$$\alpha \cdot A + \alpha \cdot B + \beta \cdot C = \{ \alpha^\varnothing, \alpha^{\alpha\beta}\alpha^\varnothing, \ldots, \beta^\beta \beta^\beta \ldots,$$
$$\alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots, \alpha^{\alpha\beta}\alpha^{\alpha\beta}\beta^\beta\beta^\beta \ldots, \ldots \}$$

6. Conclusion

In this paper, we have pointed out the importance of what is permissible (over what is possible) for a modeled system. In addition, we showed that for any transition system (used as model, see section 2) there can be constructed an outer transition system, which is always deterministic (in terms of interpretations) and still holds the same capabilities. Moreover, we showed that the capabilities identify the states of outer transition system (that is minimized, i.e. has no equal states). It simplifies many examinations of a modeled system (it also allowed us to define our algebraic structure) and it will be useful even in the further theoretical work.

Further, we discussed the equivalence of systems and examined properties of bisimulation equivalence in terms of experimenting scenarios. After that, we discovered that bisimulation is too strong for lighted-button scenario (it simply distinguishes systems that are equal in terms of this scenario). The comparing of particular capabilities (unlike bisimulation) seems to correspond to the lighted-button scenario. Hence, we are going to proof it in the further work.

In our previous work [6], the simple calculus has been introduced. Although this calculus overcome

some limitations of CCS calculus, there is an issue (in the equivalence of states), which prevented us from developing an algebraic structure. In this paper, we improved the notion of capability and showed that it is possible to introduce a proper algebraic structure over capabilities (although it is only a preview of the intended one). Our algebraic structure has two main differences from CCS calculus. First, it is simply an algebraic structure (e.g., it has just one usual kind of equation, which can replace several kinds from CCS calculus). Second, all constants and expressions have representers (i.e. sets Π^{ψ} in our case). At least, both differences simplify theoretical reasoning and allow us to utilize knowledge of abstract algebra.

7. References

[1] Baier, C., and J.P. Katoen, *Principles of Model Checking*, MIT Press, Cambridge, Massachusetts, 2008.

[2] B. Bloom, and A.R. Meyer, "Experimenting with process equivalence", *Theoretical Computer Science*, Volume 101, Issue 2, Elsevier, Essex, 1992, pp. 223-237.

[3] J. Jürjens, E.B. Fernandez, R.B. France, B. Rumpe, and C.L. Heitmeyer, "Critical Systems Development Using Modeling Languages (CSDUML-04): Current Developments and Future Challenges (Report on the Third International Workshop)", *Lecture Notes in Computer Science*, Volume 3297, Springer, Berlin, 2005, pp. 76-84.

[4] Milner, R., *Communication and Concurrency*, Prentice Hall, Harlow, England, 1989.

[5] R. Milner, J. Parrow, and D. Walker, "A Calculus of Mobile Processes - I", *Information and Computation*, Volume 100, LFCS, University of Edinburgh, 1992, pp. 1-40.

[6] P. Pinkas, and J. Klečková, "Modeling of Capabilities of a System in Algebraic Manner", *1st International Conference on Networked Digital Technologies (NDT 2009)*, Ostrava, Czech Republic, 2009, pp. 458-463.

[7] Roscoe, A.W., *The Theory and Practice of Concurrency*, Prentice Hall, Upper Saddle River, New Jersey, 1997. Revised 2005.