KYBERNETIKA — VOLUME 13 (1977), NUMBER 3

Mathematical Theory of Static Systems

EVŽEN KINDLER

The paper contains elements of a theory of static systems, based on the common set theory. It is the first part of a general theory of simulation which is under investigation and which could reflect exact properties of all kinds of simulation, including facilities built in various simulation languages for discrete, continuous and combined simulation. The outline of the theory has been presented in [5] and [6].

1. INTRODUCTION

The actual practice of simulation – namely of computer simulation – is as common at present days that the general method of it demands to obtain a suitable exact basis. One must pay attention to the fact that simulation includes various attempts to the investigated reality and to the modelling tools. Digital simulation has had use of many fruits of modern computer science, namely of programming languages: besides a large number of simulation languages of various properties and beside simulation packages there are universal programming languages of the third generation where simulation carries or has carried excellent facilities as in SIMULA 67. All these tools have been offered to the people of computer profession and thus the analytical and synthetical rules built in the programming languages have represented a suitable basis for the intended theory. Let us pay attention to programming languages – as they are offered to any user for contact with computers – bring categories of thinking which are new but common for a great part of the contemporary civilization. Beside digital simulation there is analogue and hybrid one which has not carried important categories of thinking but which should be reflected by the same theory. The same fact is true for other types of simulation used in various special branches (e.g. hydrodynamic simulation): there are certain techniques using non-computer models which nevertheless satisfy all the other demands which experts express to simulation models. The definition of simulation presented in [1] and transferred into [2] and [3] has formed a good basis for understanding simulation and one can hope it will be valid for a long future, similary as it has been valid since 10 years. It tells that simulation is the technique of replacing a dynamic system by a model, in order to gain information about the system through experiments with the model. The other presented definition of simulation are either less exact or concern aspects which is not essential for simulation: a clever philosopher can find an example which either satisfies such a definition and is evidently not a case of simulation or is evidently a case of simulation and does not satisfy the definition.

Nevertheless, the presented definition needs to specify the terms of system and model so that they exactly reflect not only the common contents accepted in practice by simulation professionals but also the richness of simulation facilities fixed in simulation programming languages.

Simulation, dynamic systems and models are general conceptions of systems theory because they can be reflected in any object, independently on its physical nature. The basic notion is system, but its known definitions have not been sufficient: that by Zadeh, based on finite automata (see e.g. [4], p. 3), cannot reflectcontinuous systems while the classical definition of dynamic system, used by specialists e.g. in regulation, based on differential equations (see e.g. [4], p. 4) loses its importance in case the fruits of differential equations cannot be applied; it is not only in case of singularities and discrete systems but also in case of variable number of system components. The last phenomenon is also the second reason for which the definition by Zadeh is not suitable: it is oriented for systems as global units while the richness of simulation (which should be reflected by richness of theorems in the corresponding theory) concerns mainly the inner structure of investigated systems and their models. Against other definitions of systems (see e.g. a lot of them presented in [7]) similar objections can be formulated.

Thus it is necessary to define appropriately the conceptions of system and of simulation model; the last conception is a relation between two systems, satisfying certain conditions. An outline of the whole theory has been presented in [5] and [6]: the theory has four main parts: theory of static systems, theory of static models, theory of dynamic systems and theory of dynamic and simulation models. The theory of static systems must be constructed as the first phase as their results must be for disposal at the theory of dynamic systems and that of static models: states of dynamic systems are static ones and static models are relations between static systems. All three theories are necessary if building that of dynamic models; the simulation models are special cases of them. In the present paper the first theory is presented in details.

As the definitions are rather complicated their english versions are completed or replaced by corresponding logical formulas. Similar practice is used to be applied in theorems, lemmas and corollaries. We shall use symbols $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\equiv$ for logical operations of negation, conjunction, disjunction, implication and equivalence; the last one is used also for the definition of predicates; the priority is decreasing with the order in the presented list. $(\ldots)$, $\exists\ldots$, $\exists!\ldots$ and $\exists^1$ are quantifiers "for every", "there is", "there is exactly one" and "there is maximally one" respectively. $\cap$, $\cup$ and $-$ are used for set intersection, sum and difference, $\in$, $\notin$, $\subseteq$ and $\subset$ for usual set relations; the last means "proper subset". The expression $\bigcup_{P \in Q} P$ or similar ones are used for a set sum of a greater number of sets, including that if Q is empty then the sum is also empty. $\times$ is used for Cartesian product, $\{f(x) \mid P(x)\}$ for a set of all $f(x)$ where x satisfies $P(x)$ and $\langle a_1, a_2, \ldots, a_n \rangle$ means an ordered n-tuplet. If f is a function then ***domain***(f) is its domain, ***range***(f) is the set of all its values and if $P \subseteq$ ***domain***(f) then f/P is f partialized to P. Empty function – with empty domain and range – is admitted.

The predicates and functions defined generally in the theory are identified by acronymes of the corresponding terms. They are printed in bold italics sans serif. As the theory is built in three levels of its hierarchy (attributes, classes, systems) we respect the following rule: the functions and the predicates concerning the attributes are identified in minuscules, those concerning the classes have only the first character in their identifier as a capital while the other ones are minuscules, the identifiers of functions and predicates concerning the systems and quasisystems are composed only of capital letters. Such a system enables to use suitable mnemonic terms for the whole hierarchy: the danger of misunderstandings is eliminated by different identifying in logical formulas. $\{a, b, \ldots\}$ has the usual meaning of the set with elements $a, b, \ldots$

2. STATIC ATTRIBUTES

We need to consider several special sets called *standard*. One of them, called $\mathscr{C}$, contains letters, digits and various signs. It has a subset $\mathscr{S}$. In the present paper it is sufficient that $\mathscr{S}$ contains two characters: point and colon. Another standard set is $\mathscr{T}$ which contains all nonempty finite sequences of the elements of $\mathscr{C}$. It has a subset $\mathscr{I}$ containing sequences which have no elements of $\mathscr{S}$. The *length* ***ln***(x) of any element x of $\mathscr{T}$ is the number of occurrences of elements of $\mathscr{S}$ in it. Although we can admit limitations to $\mathscr{I}$ which are respected for identifiers in various programming languages, the presented theory does not ask any of them. Similarly the presented theory does not depend on the decision about the logical contents of the term sequence and thus it does not depend on an answer whether $\mathscr{C} \subseteq \mathscr{T}$. As there is no danger of misunderstanding we can have use of the licence that if $a = \langle a_1, \ldots, a_n \rangle$ and $b = \langle b_1, \ldots, b_m \rangle$ are from $\mathscr{T}$ then $ab = \langle a_1, \ldots, a_n, b_1, \ldots, b_m \rangle$. If $b \in \mathscr{C}$ we read ab as $\langle a_1, \ldots, a_n, b \rangle$. The following lemma is useful.

Lemma 1. Let $a_i \in \mathscr{T}$, $v_i \in \mathscr{S}$ and $b_i \in \mathscr{I}$ for $i = 1, 2$. If $a_1 \neq a_2$ or $v_1 \neq v_2$ or $b_1 \neq b_2$ then $a_1 v_1 b_1 \neq a_2 v_2 b_2$.

Proof. Let a_1, a_2, b_1, b_2 contain n, m, s, t characters respectively. If $n + s \neq m + t$ the matter is evident. Let $n + s = m + t$, $a_1 v_1 b_1 = c_1$, $a_2 v_2 b_2 = c_2$. Let $a_1 \neq a_2$. If $n > m$ the $(n + 1)$-th character of c_1 is $v_1 \in \mathcal{S}$ while the $(n + 1)$-th one of c_2 is not of $\mathcal{S}$ as it is a component of b_2. Symmetrically for $m > n$. If $m = n$ there is $i \leqq m$ such that the i-th character of a_1 (and thus of c_1) is different from the i-th one of a_2 (and thus of c_2). If $b_1 \neq b_2$, we can limit our considerations to the case of $a_1 = a_2$, $n + s = m + t$. Hence $s = t$ and there must be $i \leqq s$ such that the i-th character of b_1 (and thus the $(i + n + 1)$-th one of c_1) is different from the i-th one of b_2 (and thus from the $(i + n + 1)$-th one) of c_2. If $v_1 \neq v_2$ we can limit our considerations to the case that $a_1 = a_2$ and $b_1 = b_2$. Then c_1 and c_2 differ in their $(n + 1)$-th characters.

The present theory does not depend on the choice of other standard sets but we can suppose that they could be other sets of values which can be assigned for usual variables in classical programming languages. Thus we can assume a set $\mathcal{R}$ of real numbers, $\mathcal{K}$ of integers, $\mathcal{B}$ of two boolean values **true** and **false** etc. It has no importance whether we suppose $\mathcal{K} \subset \mathcal{R}$ or not. $\mathbf{stand}(X)$ means that X is a standard set.

We consider an element **none** as not present in any standard set. Any set which does not contain **none** and which is disjoint with every standard set is called *nonstandard:*

$$\mathbf{gen}(Y) \equiv \mathbf{none} \notin Y \wedge (X)(\mathbf{stand}(X) \rightarrow X \cap Y = \emptyset)$$

The *closure* $\bar{X}$ of a nonstandard set X is defined as $X \cup \{\mathbf{none}\}$. Evidently $\mathbf{gen}(\emptyset)$.

Static quasiattribute is an ordered pair $\langle n, f \rangle$ where $n \in \mathcal{T}$ and f is a function the domain of which is a nonstandard set. If $a = \langle n, f \rangle$ is a static quasiattribute – we write it $\mathbf{qa}(a)$ – then its *name* is $\mathbf{nm}(a) = n$, its *domain* is $\mathbf{dm}(a) = \mathbf{domain}(f)$ and its *range* is $\mathbf{rn}(a) = \mathbf{range}(f)$. If $x \in \mathbf{dm}(a)$ then $a(x)$ is defined as $f(x)$; if $A \subseteq \mathbf{dm}(a)$ then a/A is defined as $\langle n, f/A \rangle$.

If the range of a is a standard set, a is called *standard static attribute*: $\mathbf{sa}(a) \equiv \exists X(\mathbf{stand}(X) \wedge \mathbf{rn}(a) \subseteq X)$; if the range of a contains **none** or elements of a nonstandard set, a is called *static pointer*: $\mathbf{pt}(a) \equiv \mathbf{gen}(\mathbf{dm}(a) - \{\mathbf{none}\})$. A static quasiattribute is called a *static attribute* if it is a static pointer or a standard static attribute: $\mathbf{at}(a) \equiv \mathbf{pt}(a) \vee \mathbf{sa}(a)$. Inasmuch as we shall not handle other attributes than static ones in the present paper, we shall use to omit the word *static* in case the meaning of the text is clear.

3. STATIC CLASSES

Static class is an ordered triplet $\langle n, P, G \rangle$ where $n \in \mathcal{T}$, P is a nonstandard set and G is a set of attributes with different names and with the same domain equal to P.

$$\mathbf{Cl}(A) \equiv \exists n \, \exists P \, \exists G(A = \langle n, P, G \rangle \wedge n \in \mathcal{T} \wedge \mathbf{gen}(P) \wedge (a)(a \in G \rightarrow$$
$$\rightarrow \mathbf{at}(a) \wedge \mathbf{dm}(a) = P \wedge (b)(b \in G \wedge a \neq b \rightarrow \mathbf{nm}(a) \neq \mathbf{nm}(b)))) \, .$$

Because of similar reasons as for static attributes we ommit the word static before the term class in the following text of this paper. Let $A = \langle n, P, G \rangle$ be a class. We define its *name* as $n = \mathbf{Nm}(A)$, its *domain* as $P = \mathbf{Dm}(A)$ and its *attribute-set* as $G = \mathbf{At}(A)$. In the following text, we will need intensively to handle with classes; for the simplicity we will use capital letters A, B, C and D only for classes. Thus $(A)\,P$ means $(A)(\mathbf{Cl}(A) \rightarrow P)$, $\exists B(P)$ means $\exists B(\mathbf{Cl}(B) \wedge P)$ etc. We say that a class A is *proper* if its domain contains at least one element, that it is *rich* if its attribute-set is nonempty and that it is *rich by pointers* if its attribute-set contains some pointer: $\mathbf{Pr}(A) \equiv \mathbf{Dm}(A) \neq \emptyset$, $\mathbf{Rc}(A) \equiv \mathbf{At}(A) \neq \emptyset$, $\mathbf{Rcp}(A) \equiv \exists a(a \in \mathbf{At}(A) \wedge \mathbf{pt}(a))$.

4. STATIC QUASISYSTEMS AND SYSTEMS

Static quasisystem is a set of static classes. Let Σ be a static quasisystem. We write $\mathbf{QS}(\Sigma)$ and define its domain $\mathbf{DM}(\Sigma)$ as $\bigcup_{A \in \Sigma} \mathbf{Dm}(A)$ and its attribute-set $\mathbf{AT}(\Sigma)$ as $\bigcup_{A \in \Sigma} \mathbf{At}(A)$. In the present paper we shall ommit the word static in case of static quasisystems. In the following text, Σ, Σ_1 etc. mean only static quasisystems. In case our consideration concern only one static quasisystem Σ, we shall ommit to express explicitely the membership of classes. Thus $(A)\,P$ means $(A)(A \in \Sigma \rightarrow P)$, $\exists A(P)$ means $\exists A(A \in \Sigma \wedge P)$ etc.

Σ is called *proper* if all its classes are proper: $\mathbf{PR}(\Sigma) \equiv (A)\,\mathbf{Pr}(A)$. Similarly, we define predicates *rich* and *rich by pointers*: $\mathbf{RC}(\Sigma) \equiv (A)\,\mathbf{Rc}(A)$, $\mathbf{RCP}(\Sigma) \equiv \equiv (A)\,\mathbf{Rcp}(A)$. We say that Σ is a *classical simulation one* if the domains of its classes are mutually disjoint: $\mathbf{CS}(\Sigma) \equiv (A)(B)(A \neq B \rightarrow \mathbf{Dm}(A) \cap \mathbf{Dm}(B) = \emptyset)$. We say that Σ is *well-named* if there are no two equal names of static attributes in different classes: $\mathbf{WN}(\Sigma) \equiv (A)(B)(a)(b)(a \in \mathbf{At}(A) \wedge b \in \mathbf{At}(B) \wedge A \neq B \rightarrow \rightarrow \mathbf{nm}(a) \neq \mathbf{nm}(b))$. Σ is called *name-eliminating* if any two different classes differ not only by their names: $\mathbf{NE}(\Sigma) \equiv (A)(B)(A \neq B \rightarrow \mathbf{Dm}(A) \neq \mathbf{Dm}(B) \vee \mathbf{At}(A) \neq \neq \mathbf{At}(B))$. We call Σ *fictive name one* if all its classes have identical names: $\mathbf{FN}(\Sigma) \equiv \equiv (A)(B)\,\mathbf{Nm}(A) = \mathbf{Nm}(B)$. Σ is called *SIMULA-67-named* if any two different classes have different names: $\mathbf{S67}(\Sigma) \equiv (A)(B)(\mathbf{Nm}(A) = \mathbf{Nm}(B) \rightarrow A = B)$.

Let $\leqq$ be a binary relation on Σ. We write $\mathbf{H}(\Sigma, \leqq)$ and say Σ is *hierarchical according to* $\leqq$ if the following conditions are satisfied:

(4.1) reflexivity: $(A)\,A \leqq A$,

(4.2) antisymmetry: $(A)(B)(A \leqq B \wedge B \leqq A \rightarrow A = B)$,

(4.3) transitivity: $(A)(B)(C)(A \leqq B \wedge B \leqq C \rightarrow A \leqq C)$,

(4.4) tree: $(A)(B)(C)(A \leqq C \wedge A \leqq B \rightarrow C \leqq B \vee B \leqq C)$,

(4.5) inclusion: $(A)(B)(A \leqq B \rightarrow \mathbf{Dm}(A) \subseteq \mathbf{Dm}(B))$,

(4.6) inverse inclusion: $(A)(B)(\mathbf{Dm}(A) \cap \mathbf{Dm}(B) \neq \emptyset \rightarrow A \leqq B \vee B \leqq A)$.

Static quasisystem is called *static system* if all values of its pointers are in the closure of its domain:

$$\mathbf{SS}(\Sigma) \equiv \mathbf{QS}(\Sigma) \wedge (a)(a \in \mathbf{AT}(\Sigma) \wedge \mathbf{pt}(a) \rightarrow \mathbf{rn}(a) \subseteq \overline{\mathbf{DM}(\Sigma)}) .$$

5. PROPERTIES OF STATIC SYSTEMS AND QUASISYSTEMS

In this section there are some consequences of the presented definitions. They can be formulated and proved for quasisystems though their importance is greater for static systems.

Theorem 1. Any of the following conditions implies $\mathbf{NE}(\Sigma)$:

(5.1) $\mathbf{PR}(\Sigma) \wedge \mathbf{CS}(\Sigma)$;
(5.2) $\mathbf{RC}(\Sigma) \wedge (A)(B)(A \neq B \rightarrow \mathbf{At}(A) \cap \mathbf{At}(B) = \emptyset)$;
(5.3) $\mathbf{RC}(\Sigma) \wedge \mathbf{WN}(\Sigma)$;
(5.4) $\mathbf{FN}(\Sigma)$.

Proof. Let A, B be any two different classes.

(5.1): $\mathbf{CS}(\Sigma) \rightarrow \mathbf{Dm}(A) \cap \mathbf{Dm}(B) = \emptyset$. Since $\mathbf{Pr}(A)$, $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$.

(5.2): similarly as (5.1) but instead of $\mathbf{Dm}$ we use $\mathbf{At}$.

(5.3): it follows from (5.2) as according to the definition of $\mathbf{WN}$ the second condition of (5.2) is satisfied.

(5.4): as $\mathbf{Nm}(A) = \mathbf{Nm}(B)$, there is $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$ or $\mathbf{At}(A) \neq \mathbf{At}(B)$.

Theorem 2. Let $\mathbf{RC}(\Sigma)$. If $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$ then $\mathbf{At}(A) \neq \mathbf{At}(B)$.

Proof. $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$ implies that every pair of $a \in \mathbf{At}(A)$ and $b \in \mathbf{At}(B)$ contains different elements as $\mathbf{fn}(a)$ and $\mathbf{fn}(b)$ are defined on different domains. As $\mathbf{Rc}(A)$ and $\mathbf{Rc}(B)$, such a pair can be formed and thus $\mathbf{At}(A) \neq \mathbf{At}(B)$.

Theorem 3. Let $\mathbf{H}(\Sigma, \leqq)$, A be a propre static class. Then $\mathbf{Dm}(A) \subset \mathbf{Dm}(B)$ implies $A \leqq B$, $\mathbf{Dm}(A) = \mathbf{Dm}(B)$ implies $A \leqq B \vee B \leqq A$.

Proof. $\mathbf{Dm}(A) \subset \mathbf{Dm}(B)$ implies $A \neq B$ and $\mathbf{Dm}(A) \cap \mathbf{Dm}(B) \neq \emptyset$ because $\mathbf{Pr}(A)$. According to (4.6) it is $A \leqq B$ or $B \leqq A$. $B \leqq A$ would imply $\mathbf{Dm}(A) \supseteq \mathbf{Dm}(B)$ because of (4.5) which is in contradiction with $\mathbf{Dm}(A) \subset \mathbf{Dm}(B)$ and therefore $A \leqq B$. The second statement follows from (4.6) as $\mathbf{Dm}(A) = \mathbf{Dm}(B)$ and $\mathbf{Pr}(A)$ imply $\mathbf{Dm}(A) \cap \mathbf{Dm}(B) \neq \emptyset$.

Theorem 4. Let Σ be a classical simulation quasisystem or a well-named one, x be an element of its domain and k be any text. Then $\exists^1 a(a \in \mathbf{AT}(\Sigma) \wedge \mathbf{nm}(a) = k \wedge x \in \mathbf{dm}(a))$.

Proof. Let $a \in \mathbf{AT}(\Sigma)$, $b \in \mathbf{AT}(\Sigma)$, $\mathbf{nm}(a) = \mathbf{nm}(b) = k$. According to the definition of static class, there is no $A \in \Sigma$ such that a and b could be in its attribute-set if they were not equal. But if a is in the attribute set of another class than that in the attribute set of which b is, then in case of $\mathbf{WN}(\Sigma)$ they must differ by their names and in case of $\mathbf{CS}(\Sigma)$ their domains are disjoint and thus x cannot be in both of them.

Theorem 5. Let $\mathbf{H}(\Sigma, \leqq)$, $k \in \mathscr{T}$ and $A \in \Sigma$. Then $\exists^1 B(B \in \Sigma \wedge A \leqq B \wedge \exists a(a \in \in \mathbf{At}(B) \wedge k = \mathbf{nm}(a) \wedge (C)(C \in \Sigma \wedge A \leqq C \wedge C \leqq B \wedge C \neq B \rightarrow (b)(b \in \mathbf{At}(C) \rightarrow \rightarrow k \neq \mathbf{nm}(b)))))$.

Proof. Let B and D have the properties expressed for B in the theorem. $A \leqq B$ and $A \leqq D$ and thus $B \leqq D$ or $D \leqq B$ according to (4.4). For the symmetry we can limit our considerations to $D \leqq B$. D satisfies all conditions expressed for C at the left hand side of the implication excepting $C \neq B$, but does not satisfy the statement at the right hand side of the same implication. Thus $D = B$ must hold.

6. INCLUSIONS

Let $\mathbf{QS}(\Sigma)$ and $\mathbf{QS}(\Sigma_1)$. We say that Σ_1 is a *static subquasisystem* of Σ and write $\mathbf{SB}(\Sigma_1, \Sigma)$ if there is a one-one mapping f of Σ_1 into Σ such that for any $A \in \Sigma_1$ the following conditions are satisfied:

(6.1) $\mathbf{Nm}(A) = \mathbf{Nm}(f(A))$;
(6.2) $\mathbf{Dm}(A) \subseteq \mathbf{Dm}(f(A))$;
(6.3) $(a)(a \in \mathbf{At}(A) \rightarrow \exists b(b \in \mathbf{At}(f(A)) \wedge a = b/\mathbf{Dm}(A)))$.

From the definition of $\mathbf{DM}(\Sigma)$ and from the postulate that two different attributes of the same static class cannot have identical names, the following corollary is implied:

Corollary. $\mathbf{SB}(\Sigma_1, \Sigma) \rightarrow \mathbf{DM}(\Sigma_1) \subseteq \mathbf{DM}(\Sigma)$; for any $A \in \Sigma_1$ and for any $a \in \mathbf{At}(A)$ there exist just one attribute b of (6.3).

Theorem 6. Let $\mathbf{SB}(\Sigma_1, \Sigma)$. If any of the predicates $\mathbf{CS}$, $\mathbf{WN}$, $\mathbf{FN}$ and $\mathbf{S67}$ is valid for Σ, the same is valid for Σ_1.

Proof. Let us suppose that f has the properties of the definition of $\mathbf{SB}$ and that A and B are different classes of Σ_1. Then $f(A) \neq f(B)$.

Case $\mathbf{CS}$: $f(A) \neq f(B)$ implies $\mathbf{Dm}(f(A)) \cap \mathbf{Dm}(f(B)) = \emptyset$; as $\mathbf{Dm}(A) \subseteq \mathbf{Dm}(f(A))$ and similarly for b, $\mathbf{Dm}(A) \cap \mathbf{Dm}(B) = \emptyset$.

Case $\mathbf{WN}$: let $a \in \mathbf{At}(A)$, $b \in \mathbf{At}(B)$. (6.3) implies that there are $a_1 \in \mathbf{At}(f(A))$ and $b_1 \in \mathbf{At}(f(B))$ such that $\mathbf{nm}(a_1) = \mathbf{nm}(a)$ and $\mathbf{nm}(b_1) = \mathbf{nm}(b)$. From $f(A) \neq f(B)$ and $\mathbf{WN}(\Sigma)$ it follows that $\mathbf{nm}(a) = \mathbf{nm}(a_1) \neq \mathbf{nm}(b_1) = \mathbf{nm}(b)$.

Case **FN**: because of (6.1) and the definition of **NE**, $\mathbf{Nm}(A) = \mathbf{Nm}(f(A)) = \mathbf{Nm}(f(B)) = \mathbf{Nm}(B)$.

Case **S67**: $f(A) \neq f(B)$ implies $\mathbf{Nm}(f(A)) \neq \mathbf{Nm}(f(B))$. Thus $\mathbf{Nm}(A) \neq \mathbf{Nm}(B)$ as $\mathbf{Nm}(A) = \mathbf{Nm}(f(A))$, $\mathbf{Nm}(B) = \mathbf{Nm}(f(B))$.

Remark. Similar theorems do not hold for **RC**, **RCP**, **PR** and **NE** but we can prove other ones:

Theorem 7. Let $\mathbf{SB}(\Sigma_1, \Sigma)$. Let any mapping f satisfying conditions (6.1), (6.2) and (6.3) be onto Σ. If any of the predicates **RC**, **RCP**, **PR**, **FN** and **S67** is valid for Σ_1 then the same is valid for Σ.

Proof. As f maps Σ_1 onto Σ, there exists its inverse mapping g. Let A, B be any static classes of Σ.

Case **RC**: $\exists a(a \in \mathbf{At}(g(A)))$ implies that $\exists b(b \in \mathbf{At}(A))$ and thus $\mathbf{Rc}(A)$.

Case **RCP**: we continue the last consideration: $a = b/\mathbf{Dm}(g(A)) = b/\mathbf{dm}(a)$; thus $\mathbf{rn}(a) \subseteq \mathbf{rn}(b)$ and $\mathbf{pt}(a) \rightarrow \mathbf{pt}(b)$.

Case **PR**: $\mathbf{Dm}(g(A)) \neq \emptyset$; $\mathbf{Dm}(g(A)) \subseteq \mathbf{Dm}(A)$ thus implies $\mathbf{Dm}(A) \neq \emptyset$.

Case **FN**: as always $\mathbf{Nm}(g(A)) = \mathbf{Nm}(g(B))$, according to (6.1) $\mathbf{Nm}(A) = \mathbf{Nm}(B)$.

Case **S67**: as $A \neq B \rightarrow \mathbf{Nm}(g(A)) \neq \mathbf{Nm}(g(B))$, according to (6.1) $\mathbf{Nm}(A) \neq \mathbf{Nm}(B)$.

Theorem 8. Let $\mathbf{SB}(\Sigma_1, \Sigma)$. Any of the conditions $\mathbf{CS}(\Sigma) \wedge \mathbf{PR}(\Sigma_1)$, $\mathbf{S67}(\Sigma)$, $\mathbf{WN}(\Sigma) \wedge \mathbf{RC}(\Sigma_1)$ is sufficient that the mapping f satisfying conditions (6.1), (6.2) and (6.3) is exactly one.

Proof. Let us consider any class $A \in \Sigma_1$ and any two mappings f, g satisfying the mentioned conditions, let $B = f(A)$, $C = g(A)$.

$\mathbf{PR}(\Sigma_1)$ implies $\mathbf{Dm}(A) \neq \emptyset$, $\mathbf{CS}(\Sigma)$ implies that $B \neq C \rightarrow \mathbf{Dm}(B) \cap \mathbf{Dm}(C) = \emptyset$. As $\emptyset \neq \mathbf{Dm}(A) \subseteq \mathbf{Dm}(B) \cap \mathbf{Dm}(C)$ according to (6.2), B must be identical with C.

$\mathbf{S67}(\Sigma)$ implies $\mathbf{Nm}(B) \neq \mathbf{Nm}(C)$ in case $B \neq C$ but according to (6.1) $\mathbf{Nm}(A) = \mathbf{Nm}(B) = \mathbf{Nm}(C)$.

$\mathbf{WN}(\Sigma) \wedge \mathbf{RC}(\Sigma_1)$ implies that there is a static attribute $a \in \mathbf{At}(A)$. From (6.3), it follows that there is $b \in \mathbf{At}(B)$ and $c \in \mathbf{At}(C)$ such that $\mathbf{nm}(b) = \mathbf{nm}(c) = \mathbf{nm}(a)$; since in well-named quasisystems the attributes from different classes have not the same names, B must be identical with C.

Theorem 9. Let $\mathbf{SB}(\Sigma_1, \Sigma)$ and $\mathbf{H}(\Sigma, \leqq)$. Let f be a mapping satisfying conditions (6.1), (6.2) and (6.3) and be one-one mapping similarly as in the definition of **SB**. We can therefore define a binary relation R_f on Σ_1 as $R_f(A, B) \equiv f(A) \leqq f(B)$. R_f satisfies conditions (4.1) to (4.4) and (4.6) of the definition of hierarchical quasisystem. If moreover any pair of static classes C and D of $f(\Sigma_1)$ satisfy the implication $\mathbf{Dm}(C) \subseteq \mathbf{Dm}(D) \rightarrow \mathbf{Dm}(f^{-1}(C)) \subseteq \mathbf{Dm}(f^{-1}(D))$ then $\mathbf{H}(\Sigma_1, R_f)$.

Proof. $f(A) \leqq f(A)$ implies $R_f(A, A)$. $R_f(A, B) \wedge R_f(B, A)$ implies $f(A) \leqq \leqq f(B) \wedge f(B) \leqq f(A)$; according to (4.2) it implies $f(A) = f(B)$ and thus $A = B$ as f is one-one. $R_f(A, B) \wedge R_f(B, C)$ implies $f(A) \leqq f(B) \wedge f(B) \leqq f(C)$; according to (4.3), $f(A) \leqq f(C)$, and thus $R_f(A, C)$. $R_f(A, C) \wedge R_f(A, B)$ implies $f(A) \leqq \leqq f(C) \wedge f(A) \leqq f(B)$; according to (4.4), $f(C) \leqq f(B) \vee f(B) \leqq f(C)$ and thus $R_f(C, B) \vee R_f(B, C)$. $\mathbf{Dm}(A) \cap \mathbf{Dm}(B) \neq \emptyset$ implies $\mathbf{Dm}(f(A)) \cap \mathbf{Dm}(f(B)) \neq \emptyset$ because of (6.2). For (4.6), $f(A) \leqq f(B) \vee f(B) \leqq f(A)$ and thus $R_f(A, B) \vee R_f(B, A)$. Let $R_f(A, B)$; then $f(A) \leqq f(B)$ and thus $\mathbf{Dm}(f(A)) \subseteq \mathbf{Dm}(f(B))$ according to (4.5). If the last condition of Theorem 9 is satisfied then also $\mathbf{Dm}(A) \subseteq \mathbf{Dm}(B)$ and thus R_f satisfies all properties of the definition of $\boldsymbol{H}$.

Let $\mathbf{SB}(\Sigma_1, \Sigma)$. We say that Σ_1 is a *static subquasisystem of Σ with identical domains* and write $\mathbf{SBD}(\Sigma_1 \Sigma)$ if every mapping f satisfying the properties of the definition of $\mathbf{SB}$ is a mapping onto Σ and $\mathbf{Dm}(f(A)) = \mathbf{Dm}(A)$ is valid for it and for every $A \in \Sigma$. Let us mention that in this case we can apply Theorem 7 and that from Theorem 8 the following corollary follows:

Corollary. Let $\mathbf{SB}(\Sigma_1, \Sigma)$ and let there be a one-one mapping f of Σ_1 onto Σ satisfying (6.1), (6.2) and (6.3) and $\mathbf{Dm}(f(A)) = \mathbf{Dm}(A)$ for every $A \in \Sigma$. Any condition of $\mathbf{CS}(\Sigma) \wedge \mathbf{PR}(\Sigma_1)$, $\mathbf{S67}(\Sigma)$ or $\mathbf{WN}(\Sigma) \wedge \mathbf{RC}(\Sigma_1)$ is then sufficient that $\mathbf{SBD}(\Sigma_1, \Sigma)$.

If the condition of the last Corollary is satisfied, we write $\mathbf{SBDW}(\Sigma_1, \Sigma)$ and say that Σ_1 is a *static subquasisystem of Σ with identical domain in a week sense*. Let us note that in case of f used in the last Corollary or in the definition of $\mathbf{SBD}$ we can modify (6.3) as $\mathbf{At}(A) \subseteq \mathbf{At}(f(A))$, and of course (6.2) as $\mathbf{Dm}(A) = \mathbf{Dm}(f(A))$. Evidently $\mathbf{SBD}(\Sigma_1, \Sigma) \rightarrow \mathbf{SBDW}(\Sigma_1, \Sigma)$ but the inverse implication is valid not generally (the conditions of the last Corollary can ensure it in certain cases).

Theorem 10. Let $\mathbf{SBDW}(\Sigma_1, \Sigma)$. If P is $\mathbf{PR}$, $\mathbf{CS}$, $\mathbf{FN}$ or $\mathbf{S67}$ then $P(\Sigma) \equiv P(\Sigma_1)$. If P is $\mathbf{RC}$ or $\mathbf{RCP}$ then $P(\Sigma_1) \rightarrow P(\Sigma)$. $\mathbf{WN}(\Sigma) \rightarrow \mathbf{WN}(\Sigma_1)$.

Proof. As we can have use of Theorems 6 and 7 we must take into account only the following implications: $\mathbf{PR}(\Sigma) \rightarrow \mathbf{PR}(\Sigma_1)$, and $\mathbf{CS}(\Sigma_1) \rightarrow \mathbf{CS}(\Sigma)$. They follow from the above mentioned modification of (6.2) and from the condition that f is a one-one mapping onto.

Theorem 11. Let $\leqq$ be a relation defined on Σ, f have the properties expressed in the last corollary and R_f be defined on Σ_1 similarly an in Theorem 9. Then $\mathbf{H}(\Sigma, \leqq) \equiv \mathbf{H}(\Sigma_1, R_f)$. (Evidently $\mathbf{SBDW}(\Sigma_1, \Sigma)$ in that case.)

Proof. $\mathbf{H}(\Sigma, \leqq) \rightarrow \mathbf{H}(\Sigma_1, R_f)$ follows from Theorem 9. The proof of the inverse statement can be easily performed similarly as the proof of Theorem 9 if we use f^{-1} instead of $f : f^{-1}$ is fully defined and one-one mapping.

We say that Σ_1 is a *static subsystem* of Σ if Σ and Σ_1 are static systems and Σ_1 is a static subquasisystem of $\Sigma : \mathbf{SBS}(\Sigma_1, \Sigma) \equiv \mathbf{SB}(\Sigma_1, \Sigma) \wedge \mathbf{SS}(\Sigma_1) \wedge \mathbf{SS}(\Sigma)$. Evidently, if P is $\mathbf{SB}$, $\mathbf{SBS}$ or $\mathbf{SBDW}$ then $P(\Sigma_1, \Sigma) \wedge P(\Sigma, \Sigma_2) \rightarrow P(\Sigma_1, \Sigma_2)$.

Let $\langle m, f \rangle = a$ be a static pointer of a static system Σ. We say that it is *qualified* into M where $M \subseteq \Sigma$ if $\mathbf{rn}(a) \subseteq \overline{\bigcup_{A \in M} \mathbf{Dm}(A)}$, eventually *strictly qualified* into M if $\mathbf{rn}(a) \subseteq \bigcup_{A \in M} \mathbf{Dm}(A)$. We write $\mathbf{q}(a, M)$ eventually $\mathbf{sq}(a, M)$. Let $\mathbf{q}(a, \{A\})$ and $b = \langle n, g \rangle \in \mathbf{At}(A)$. If $\mathbf{pt}(b)$ we define *junction* $\mathbf{jn}(a, b)$ of a and b as a static pointer the name of which is $m : n$ and the function forming its second component is defined for $x \in \mathbf{dm}(a)$ as $g(f(x))$ in case $f(x) \neq$ **none** and as **none** otherwise. If $\mathbf{sq}(a, \{A\})$ we define *strict junction* $\mathbf{sjn}(a, b)$ of a and b as a static attribute the name of which is $m \,.\, n$ and the function forming its second component is defined for $x \in \mathbf{dm}(a)$ as $g(f(x))$.

In case $\mathbf{jn}(a, b)$ eventually $\mathbf{sjn}(a, b)$ are defined, $\mathbf{pt}(\mathbf{jn}(a, b))$, $\mathbf{pt}(\mathbf{sjn}(a, b)) \equiv \mathbf{pt}(b)$, $\mathbf{sa}(\mathbf{sjn}(a, b)) \equiv \mathbf{sa}(b)$, $\mathbf{dm}(\mathbf{jn}(a, b)) = \mathbf{dm}(a)$, $\mathbf{dm}(\mathbf{sjn}(a, b)) = \mathbf{dm}(a)$, $\mathbf{rn}(\mathbf{jn}(a, b)) \subseteq \overline{\mathbf{rn}(b)}$ and $\mathbf{rn}(\mathbf{sjn}(a, b)) \subseteq \mathbf{rn}(b)$. Let us mention that $\mathbf{nm}(\mathbf{jn}(a, b)) \neq \mathbf{nm}(\mathbf{sjn}(a, b))$ even if $\mathbf{sq}(a, \{\mathbf{dm}(b)\})$, and thus $\mathbf{jn}(a, b) \neq \mathbf{sjn}(a, b)$ if both exist.

We call a static quasisystem *simple* if the names of all static attributes of its attribute set are elements of $\mathscr{I}$. In the further considerations, let Σ be a simple well-named static system.

Theorem 12. Let $A \in \Sigma$ and the sequences $\{V_i^k\}_{i=1}^{\infty}$, $k = 1, 2$ are defined recursively: $V_1^1 = \mathbf{At}(A)$, $V_1^2 = \emptyset$,

$$V_{i+1}^1 = \{a \mid \exists b\, \exists c (b \in \mathbf{AT}(\Sigma) \wedge c \in V_i^1 \cup V_i^2 \wedge a = \mathbf{sjn}(c, b))\},$$

$$V_{i+1}^2 = \{a \mid \exists b\, \exists c (b \in \mathbf{AT}(\Sigma) \wedge c \in V_i^1 \dot{\cup} V_i^2 \wedge a = \mathbf{jn}(c, b))\}.$$

Let $X = \bigcup_{i=1}^{\infty} (V_i^1 \cup V_1^2)$, $B = \langle \mathbf{Nm}(A), \mathbf{Dm}(A), X \rangle$. Then the following statements are valid:

(7.1) if $j > 0$, $a \in V_j^1$ and $k = \mathbf{nm}(a)$ then $k = n.m$ where $m \in \mathscr{I}$ and $n \in \mathscr{T}$; in case $j = 0$, $k \in \mathscr{I}$;
(7.2) if $a \in V_j^2$ then $\mathbf{nm}(a) = n : m$ where $m \in \mathscr{I}$ and $n \in \mathscr{T}$;
(7.3) if $a \in V_j^1 \cup V_j^2$ then $\mathbf{ln}(\mathbf{nm}(a)) = j - 1$; $\mathbf{dm}(a) = \mathbf{Dm}(A)$;
(7.4) if $a \in V_i^1$ and $b \in V_j^2$ then $\mathbf{nm}(a) \neq \mathbf{nm}(b)$;
(7.5) if $i \neq j$, $a \in V_i^k$ and $b \in V_j^k$ then $\mathbf{nm}(a) \neq \mathbf{nm}(b)$;
(7.6) $V_i^1 \cap V_j^2 = \emptyset$; if $i \neq j$ then $V_i^k \cap V_j^k = \emptyset$;
(7.7) if $a \in X$ is a pointer then $\mathbf{rn}(a) \subseteq \overline{\mathbf{DM}(\Sigma)}$;
(7.8) if $a \in V_i^k$, c is a character contained in $\mathbf{nm}(a)$ and $c \in \mathscr{S}$ then c is either a point or a colon;
(7.9) if $a \in X$ and $b \in X$ then $\mathbf{nm}(a) = \mathbf{nm}(b) \rightarrow a = b$;
(7.10) $\mathbf{Cl}(B)$.

Static class B constructed according to the way of the preceding theorem from A is called *enlargement* of A in Σ and written $\mathbf{En}(A, \Sigma)$.

Proof. (7.1) and (7.2) follow immediately from the definition of V_i^k. (7.3), (7.7) and (7.8) follow from the same definition by a simple induction. (7.1) and (7.2) imply (7.4), (7.3) implies (7.5). (7.4) and (7.5) imply (7.6). To prove (7.9) we can limit our considerations to a case that a and b belong to the same V_i^k because otherwise they cannot have the same names for (7.4) and (7.5). Let $a \in V_i^k$, $b \in V_i^k$. If $i = 1$ then $k = 1$ and (7.9) is satisfied as V_1^1 is an attribute set. Let us suppose that $i > 1$ and that the elements of V_{i-1}^1 and of V_{i-1}^2 satisfy (7.9). Let us further suppose that $\mathbf{nm}(a) = \mathbf{nm}(b)$ and that $k = 1$ (for $k = 2$ the considerations are similar). Then $a = \mathbf{sjn}(\bar{a}, \bar{\bar{a}})$, $b = \mathbf{sjn}(\bar{b}, \bar{\bar{b}})$ and thus $\mathbf{nm}(a) = \mathbf{nm}(\bar{a}) \,.\, \mathbf{nm}(\bar{\bar{a}}) = \mathbf{nm}(\bar{b}) = \mathbf{nm}(\bar{b}) \,.\, \mathbf{nm}(\bar{\bar{b}})$. According to Lemma 1, $\mathbf{nm}(\bar{a}) = \mathbf{nm}(\bar{b})$ and $\mathbf{nm}(\bar{\bar{a}}) = \mathbf{nm}(\bar{\bar{b}})$. According to the induction supposition, $\bar{a} = \bar{b}$; as Σ is a well-named system, $\mathbf{nm}(\bar{\bar{a}}) = \mathbf{nm}(\bar{\bar{b}})$ implies $\bar{\bar{a}} = \bar{\bar{b}}$. Therefore $a = \mathbf{sjn}(\bar{a}, \bar{\bar{a}}) = \mathbf{sjn}(\bar{b}, \bar{\bar{b}}) = b$. (7.10) follows immediately from the definition of B and frcm (7.9) and (7.3).

Theorem 13. The following statements hold for any $A \in \Sigma$:

(7.11) $\mathbf{At}(A) \subseteq \mathbf{At}(\mathbf{En}(A, \Sigma))$
(7.12) $\mathbf{Dm}(A) = \mathbf{Dm}(\mathbf{En}(A, \Sigma))$
(7.13) $\mathbf{Rc}(A) \to \mathbf{Rc}(\mathbf{En}(A, \Sigma))$
(7.14) $\mathbf{Rcp}(A) \to \mathbf{Rcp}(\mathbf{En}(A, \Sigma))$
(7.15) $\sim \mathbf{Rc}(A) \to \sim \mathbf{Rc}(\mathbf{En}(A, \Sigma))$
(7.16) $\sim \mathbf{Rcp}(A) \to \sim \mathbf{Rcp}(\mathbf{En}(A, \Sigma))$
(7.17) $a \in \mathbf{At}(\mathbf{En}(A, \Sigma)) \wedge \mathbf{In}(a) = 0 \to a \in \mathbf{At}(A)$
(7.18) if $A \in \Sigma$ and $B \in \Sigma$ and if $\mathbf{At}(A) = \mathbf{At}(B)$ then $\mathbf{At}(\mathbf{En}(A, \Sigma)) = \mathbf{At}(\mathbf{En}(B, \Sigma))$.

Proof. (7.11) follows immediately from the definition of X presented in the preceding theorem. (7.12) follows immediately from (7.3). (7.13) and (7.14) follow from (7.11). If a class has no pointers (if it is not rich or not rich by pointers) the sets V_1^k constructed from it as in Theorem 12 have no pointers and V_i^k for $i > 1$ are empty. Thus (7.15) and (7.16) are proved as in the considered case $\mathbf{At}(A) = \mathbf{At}(\mathbf{En}(A, \Sigma)$. (7.17) follows from (7.11) and (7.3). (7.18) follows from the definition by a simple induction.

We call $\{\mathbf{En}(A, \Sigma) \mid A \in \Sigma\}$ to be (*static*) *enlargement* of Σ and write it $\mathbf{EN}(\Sigma)$. Evidently it is a static quasisystem.

Theorem 14. $\mathbf{SS}(\mathbf{EN}(\Sigma)) \wedge \mathbf{SBS}(\Sigma, \mathbf{EN}(\Sigma)) \wedge \mathbf{SBDW}(\Sigma, \mathbf{EN}(\Sigma)) \wedge \mathbf{WN}(\mathbf{EN}(\Sigma))$.

Proof. (7.7) implies that $\mathbf{EN}(\Sigma)$ is a static system. Let us consider the mapping $f(A) = \mathbf{En}(A, \Sigma)$. According to the definition of $\mathbf{En}(A, \Sigma)$ in Theorem 12, (6.1) is satisfied. According to (7.12), (6.2) is satisfied but moreover, the condition $\mathbf{Dm}(A) = \mathbf{Dm}(\mathbf{En}(A, \Sigma))$ of the definition of $\mathbf{SBDW}$ is satisfied (see the corollary following

Theorem 9). (7.11) implies (6.3) in its modified form presented before Theorem 10. Thus $\mathbf{SBS}(\Sigma, \mathbf{EN}(\Sigma))$ and $\mathbf{SBDW}(\Sigma, \mathbf{EN}(\Sigma))$ are proved. $\mathbf{WN}(\mathbf{EN}(\Sigma))$ can be proved by induction: Let $a \in \mathbf{At}(\mathbf{En}(A, \Sigma))$, $b \in \mathbf{At}(\mathbf{En}(B, \Sigma))$. Let $\mathbf{nm}(a) = \mathbf{nm}(b)$. If $\mathbf{ln}(\mathbf{nm}(a)) = \mathbf{ln}(\mathbf{nm}(b)) = 0$, for (7.17) $a \in A$ and $b \in B$ and for $\mathbf{WN}(\Sigma)$, $A = B$. Let $\mathbf{nm}(a) = \mathbf{nm}(b)$ imply $A = B$ for any a, b with the length less than i of their names. Let us consider the static attributes of the length equal i of their names. According to (7.8), $\mathbf{nm}(a) = \mathbf{nm}(\bar{a})\, u\, \mathbf{nm}(\bar{\bar{a}})$ and $\mathbf{nm}(b) = \mathbf{nm}(\bar{b})\, v\, \mathbf{nm}(\bar{\bar{b}})$ where $\mathbf{nm}(\bar{a}) \in \mathscr{I}$ and $\mathbf{nm}(\bar{b}) \in \mathscr{I}$. According to Lemma 1, $\mathbf{nm}(\bar{a}) = \mathbf{nm}(\bar{b})$ and according to the induction supposition, $\bar{a}$ and $\bar{b}$ must belong to the attribute set of the same class. As presented in the construction of V_j^k, a and b belong to the same class as $\bar{a}$ and $\bar{b}$.

Theorem 15. Let P be any of the predicates $\mathbf{PR}$, $\mathbf{RC}$, $\mathbf{RCP}$, $\mathbf{CS}$, $\mathbf{FN}$, $\mathbf{S67}$ or $\mathbf{NE}$. Then $P(\Sigma) \equiv P(\mathbf{EN}(\Sigma))$.

Proof. Because of the Theorem 10, we must prove only several statements: $\mathbf{RC}(\mathbf{EN}(\Sigma))$ implies $\mathbf{RC}(\Sigma)$ for (7.15), $\mathbf{RCP}(\mathbf{EN}(\Sigma))$ implies $\mathbf{RCP}(\Sigma)$ for (7.16). If $\mathbf{NE}(\Sigma)$ and $\mathbf{Nm}(A) \neq \mathbf{Nm}(B)$ for two static classes of Σ, then $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$ or $\mathbf{At}(A) \neq \mathbf{At}(B)$. In the first case $\mathbf{Dm}(\mathbf{En}(A, \Sigma)) \neq \mathbf{Dm}(\mathbf{En}(B, \Sigma))$ because of (7.12) and in the second case there is an attribute $a \in (\mathbf{At}(A) - \mathbf{At}(B)) \cup (\mathbf{At}(B) - \mathbf{At}(A))$. For the symmetry we can suppose that $a \in \mathbf{At}(A) - \mathbf{At}(B)$. Because of (7.11) and (7.17) it is in $\mathbf{At}(\mathbf{En}(A, \Sigma))$ but it is not in $\mathbf{At}(\mathbf{En}(B, \Sigma))$. If $\mathbf{NE}(\Sigma)$ and $\mathbf{Nm}(\mathbf{En}(A, \Sigma)) \neq \mathbf{Nm}(\mathbf{En}(B, \Sigma))$ for two static classes $\mathbf{En}(A, \Sigma)$ and $\mathbf{En}(B, \Sigma)$ of $\mathbf{EN}(\Sigma)$ then $\mathbf{Dm}(\mathbf{En}(A, \Sigma)) \neq \mathbf{Dm}(\mathbf{En}(B, \Sigma))$ or $\mathbf{At}(\mathbf{En}(A, \Sigma)) \neq \mathbf{At}(\mathbf{En}(B, \Sigma))$. In the first case $\mathbf{Dm}(A) \neq \mathbf{Dm}(B)$ because of (7.12). In the second case, $\mathbf{At}(A) \neq \mathbf{At}(B)$ because of (7.18).

Theorem 16. Let $\mathbf{WN}(\Sigma)$, $\mathbf{H}(\Sigma, \leqq)$. Let R be a binary relation defined on $\mathbf{EN}(\Sigma)$ as $R(\mathbf{En}(A, \Sigma), \mathbf{En}(B, \Sigma)) \equiv A \leqq B$. R is defined correctly and $\mathbf{H}(\mathbf{EN}(\Sigma), R)$.

Proof. According to the proof of theorem 14, $f(A)$ defined on $\mathbf{EN}(\Sigma)$ as $\mathbf{En}(A, \Sigma)$ satisfies the properties demanded for f in Theorem 11. R is identical with R_f of the same theorem.

8. EXPLICATIONS OF INTRODUCED TERMS

The notions of static attributes, pointers, classes and systems correspond to instantaneous states of the notions which are commonly known in computer simulation under the same words. Such notions as quasiattributes and namely quasisystems have been introduced only for a better legibility of other definitions but one can see that a lot of theorems can be proved for them although one could expect their validity for the notions named without the prefix *quasi*. Definitions of various types of quasisystems correspond to certain cases which are not "singular" (as proper, rich,

rich by pointer) or to programming methodology introduced in various groups of programming languages: thus ***CS*** corresponds to conception of systems respected in various discrete event simulation languages, where one element of a system cannot be present in two classes. ***S67*** corresponds to the methodology introduced in SIMULA 67 where one element can be present in a class which is a subclass of another class: such a subclass is reflected in the present theory as another class which differs from the including one by its name and possibly by attributes (in case the subclass has declared its proper ones). ***WN*** reflects a phenomenon that in a lot of simulation languages two attributes of different classes must differ by their names: such a rule can be introduced also implicitely, by so-called qualified referencing, as we know from SIMULA 67, NEDIS, GSL and even from PL/1. Certain simulation languages do not introduce the names of classes; it is reflected by ***FN***, where the identical names of classes have no influence to their differing. ***NE*** is a certain generalization of ***FN***. Hierarchical relations on systems are reflections to SIMULA 67 (see [8]) but we can apply them also for other programming languages with hierarchical structure, which have no importance for computer simulation.

Theorems 1, 2 and 3 contain relations among types of static systems which hold in "non-singular" cases. The following 2 theorems reflect properties of attribute identification used commonly in simulation languages. The theory of subsystems has been introduced mainly for the following theory of system enlargements: this one reflects certain properties of definitions in systems. We have limited the considerations to junctions and strict junctions in the well-named static systems, although one could generalize them for other types of definitions and for other types of systems: such generalizations would exceed the task of the present introductory paper while the matter in the bounds of it can be used for basic techniques in simulation: if e.g. *first* is an attribute of a queue, pointing to the first element of it or to **none** if the queue is empty, and if *suc* is an attribute of any element which can enter in a queue, pointing to its successor or to **none** if it does not exist, then the attribute "the second element of a queue" can be expressed as the junction of *first* and *suc*.

We did not eliminate the "singular" cases directly in the definition of the static system, for the states of dynamic ones are very often singular: a state of a dynamic system can often have some static classes which are not proper. Another reason for the same matter is that if we respect advanced manners in computer model building and design (also by means of primitive simulation languages as GPSS) we can see frequent use of "singular" cases as classes which are not rich or not rich by pointers. Thus the further theory of dynamic systems would be incomplete if the states of the dynamic systems could be only proper, rich etc. It would be possible to introduce a notion of a static class without demand that two different static attributes of it must have different names. It could simplify several proofs but other theorems would be valid and the entire theory would be more illegible; we have not met any simulation language where the mentioned demand would not be satisfied.

(Received July 8, 1976.)

[1] O. - J. Dahl: Discrete event simulation languages. Norsk Regnesentral, Oslo 1966.

[2] F. Genuys (ed.): Programming languages. Academic Press, London—New York 1968.

[3] Ф. Женюи (ред.): Языки программирования. Мир, Москва 1972.

[4] P. Brumovský, J. Černý: Matematická teória systémov — mýty a skutočnosť. In: Zborník 7. sympósia SKS o aplikáciach teoretických principov kybernetiky — prehladové referáty. Bratislava 1976, 1—32.

[5] E. Kindler: Teorie simulačních modelů. In: Zborník 7. sympózia SKS o aplikáciach teoretických principov kybernetiky — 1. časť, Bratislava 1976, 84—93.

[6] E. Kindler: On the way to mathematical theory of simulation. Elektronische Informationsverarbeitung und Kybernetik 1976, 10, 497—504.

[7] G. J. Klir: Trends in general systems theory. Wiley-Interscience, New York—London—Sydney—Toronto 1972.

[8] O.-J. Dahl, B. Myhrhaug, K. Nygaard: SIMULA 67 common base language. 2. ed. Norsk Regnesentral, Oslo 1972.

PhDr. RNDr. Evžen Kindler, CSc., katedra matematické informatiky, matematicko-fyzikální fakulta Karlovy university (Department of Mathematical Informatics, Faculty of Mathematics and Physics, Charles University), Malostranské nám. 25, 118 00 Praha 1. Czechoslovakia.