Semantics of Systems of Concurrent Processes

Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

469

I. Guessarian (Ed.)

Semantics of Systems of Concurrent Processes

LITP Spring School on Theoretical Computer Science
La Roche Posay, France, April 23–27, 1990
Proceedings

Springer-Verlag
Berlin Heidelberg New York London
Paris Tokyo Hong Kong Barcelona

Volume Editor
Irène Guessarian
Université Paris VI
LITP, Tour 45–55, 4 place Jussieu
F-75252 Paris Cedex 05, France

CR Subject Classification (1987): F.3–4, D.4, C.1.2, H.2.4

ISBN 3-540-53479-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-53479-2 Springer-Verlag New York Berlin Heidelberg

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

FOREWORD

The present volume contains the proceedings of the 1990 Spring School of Theoretical Computer Science which was devoted this year to the Semantics of Concurrency. The School was organized jointly by the LITP (Laboratoire d'Informatique Théorique et Programmation, Paris) and IRIT (Institut pour la Recherche en Informatique de Toulouse). The talks were divided into two series:

- tutorial talks, which introduced the subject to neophytes;
- advanced talks, which presented a survey of recent achievements in Semantics of Concurrency.

The following is a detailed list of the tutorial talks given: Transitions systems and the semantics of communicating processes (*A. Arnold*), Process algebra: CCS and MEIJE (*G. Boudol*), Mobile processes (*R. Milner*), Testing equivalences (*M. Hennessy*), An introduction to metric semantics (*J. W. de Bakker*), Parallelism and computability (*Ph. Darondeau*), CCS and Petri nets (*U. Goltz*), Specification and verification of process properties (*J. Sifakis and A. Arnold*), Event structures (*I. Castellani*), Categories for parallelism (*U. Montanari*). Most of these talks are not contained in the present proceedings, or only very briefly covered, with references to the existing literature, because they are already very well documented in various papers.

The advanced talks, on the other hand, are contained in the present volume. These talks center around four main themes:

- Models for concurrent and distributed systems: this area includes pomsets and metric semantics (J. W. de Bakker, J. H. A. Warmerdam), event structures (G. Boudol, I. Castellani), causal trees (Ph. Darondeau, P. Degano), partial orders and languages (B. Rozoy), fixpoints and languages (I. Guessarian), trace monoids (G. Duchamp, D. Krob, P. Gastin), CCS and Petri nets (U. Goltz), categorical models (U. Montanari, S. Kasangian, A. Labella, A. Pettorossi).
- Observation and bisimulation equivalences: this part includes observational semantics and abstract data types (E. Astesiano, A. Giovini, G.Reggio), preservation of equivalences by refinements (R. J. van Glabbeek).
- Logics for concurrency: this area contains computability notions for operational specifications (Ph. Darondeau), for fairness (L. Priese), bisimulation logics (R. De Nicola, F. Vaandrager).
- Applications to distributed systems: this part covers the following: parallel languages for SIMD connection machines (L. Bougé), problems of distributed systems such as byzantine generals (J. Beauquier), or notions of clock (B. Charron).

I would like to thank the members of the program committee of the school, which consisted of *André Arnold, Joffroy Beauquier, Gérard Boudol, Philippe Darondeau, Irène Guessarian* and *Maurice Nivat*. Special thanks are due to *Maurice Nivat*, who for 20 years has consistently animated and each time given a new spirit to the spring schools. Special thanks also go to *Colette Ravinet* and *Patrick Sallé* who most efficiently managed all practical problems before and during the conference, for their perfectly equal and amiable temper throughout all events. Thanks are due also to the City Hall, the Syndicat d'Initiative of La Roche Posay, and the managers and employees of the hotel de l'Esplanade.

Thanks are due finally to the Département Sciences Physiques pour l'Ingénieur of the CNRS, the PRC Mathématiques et Informatique, and most specially (last but not least) the GRECO Communication, Concurrence, Coopération, whose financial support made this school possible. The proceedings of the last five schools have also been published in the Springer-Verlag LNCS series (Vols. 192, 242, 316, 377 and 386).

Paris, October 1990 — Irène Guessarian

TABLE OF CONTENTS

Processes as Data Types: Observational Semantics and Logic*
(Extended Abstract)

Egidio Astesiano Alessandro Giovini Gianna Reggio
Department of Mathematics
University of Genova - Italy

Introduction

We present here an attempt towards a unifying approach for the semantics of concurrency, abstracting from the particular language used for describing processes. The original motivation of this work was the integration of process specifications into the general schema of algebraic specifications of abstract data types (see [AR,AGR2] for the general approach). In this area abstraction from the language is essential. Indeed whenever some data are processes, in order to keep a reasonable level of abstraction, processes are to be specified just as special elements in some algebraic structure and moreover their semantics has to fit into the overall semantics of the specification. Now it is rather well-known that the classical notions of semantics for algebraic specifications turn out to be not adequate for expressing sensible semantics for processes. Our approach is to learn and abstract from the fundamental studies on calculi like *CCS* some basic ideas, showing how they can be lifted to a treatment not depending on the language and accomodating a variety of semantics.

Processes are here abstractly viewed as elements of observable sort in an algebraic structure (in secion 1 we briefly introduce some examples of this approach, viewing processes as algebraic transition systems; but note that this view is not essential for the following theory). In order to define a semantics we embody in the algebraic structure an observational viewpoint, obtaining what we call an *observational structure* (section 2).

Essentially an observational structure consists of an algebra equipped with

- *experiments*: possibly infinitary first order contexts for observable elements;
- a *similarity law* for experiments: a function which, given a (similarity) relation on the elements of the algebra, generates a similarity relation on experiments;
- a *propagation law* for relations: a function which propagates a (similarity) relation on the observable elements to a (similarity) relation on elements of the other sorts.

*Work partially funded by COMPASS-Esprit Basic Research Group No. 3264 and by CNR-PF-Sistemi Informatici e Calcolo Parallelo.

With each observational structure an *observational equivalence* is associated, which is an abstract version of the well-known bisimulation equivalence of [P] for transition systems. In order to explore further this correspondence, we introduce the notion of *representable observational structure*: essentially a structure where the similarity law for experiments is representable by families of patterns of experiments. The main result of the paper shows how to associate with a representable observational structure a set of *modal observational logic formulas* (section 3), such that an abstract version of the Hennessy-Milner theorem holds: two observable elements are observationally equivalent iff they satisfy the same set of such formulas.

It is shown that interesting examples (not only strong and weak, but also distributed and branching bisimulation) can be seen as instances of representable observational structures, and so we get a corresponding modal observational logic. Throughout the paper we use variations of *CCS* to illustrate ideas, definitions and applications.

The problem of a sensible generalization of the notion of bisimulation has been first tackled in [AW], where a lattice of simulation relations is defined, whose greatest element can be seen as a possible generalization of Park and Milner's notion of bisimulation in an algebraic framework; in [AGR1] a different generalization closer to the original definition is proposed; in [GR] it is shown that this generalization is indeed quite natural, and are given also sufficient conditions for the maximum observational relation to be a congruence and generate a model. Applications of the notion of generalized bisimulation to concurrency can be found in [AR] (where a family of parametric concurrent calculi integrating processes, functions and abstract data types is defined and its properties are studied) and in [AGR2] (where several examples of processes used as data types are given); while applications to the semantics of abstract data types can be found in [AGR1]. Our work, together with generalizing the Hennessy-Milner work (see [M2]) to general algebraic structures, is clearly much related to the work by De Nicola and Hennessy on testing equivalences (see [DH]), and the relationship will be partly clarified in the paper. We also feel that in the framework of observational structures it is possible to formalize and deal with the hierarchies of semantics for concurrent processes presented by Abramsky in [A]; this will be the subject of further work.

Arnold and Dicky [AD] and Ferrari and Montanari [FM] work in a similar direction to ours, aiming at a general framework for the semantics of concurrency. Their approaches are however different; they define classes of models (Φ-algebras in [AD], the $\mathcal{U}CCS$ category in [FM]) and of morphisms (quasi-saturating homomorhpisms in [AD], abstraction homomorphisms in [FM], a notion introduced in [C]) and get the notion of maximum observational equivalence via terminality. A deeper analysis of the relationship between our and their work would probably be of interest. Also, it is a research topic to be examined whether with each observational structure can be associated a category such that the observational equivalence (or, the maximum congruence contained in it) can be obtained via terminality; some preliminary investigations can be found in [GR].

In this paper we do not deal with the application of the theory of observational structures to algebraic specifications in general; for this we refer to a full paper which includes also the proofs of all the results (see [AGR3]).

1 Processes as Data Types

In this section we briefly show how processes can be formally described as data types in an algebraic style, adopting the view of *CCS* that processes are labelled transition systems; however this viewpoint is not essential to the following theory, where processes are just modelled by algebras. The examples, centered around *CCS*, will be used throughout the paper.

1.1 The algebraic framework

We briefly summarize our formal framework, which is that of *total algebras with predicates*. The basic definitions and results can be found in [GM]; here we repeat just the essential notions.

A *signature* Σ consists of a set of *sorts* (S), a family of *operation symbols* ($F = \{F_{\omega,s}\}_{\omega \in S^*, s \in S}$) and a family of *predicate symbols* ($P = \{P_\omega\}_{\omega \in S^*}$); moreover we indicate by

- $f: s_1 \times \cdots \times s_n \to s$ the fact that $f \in F_{s_1 \ldots s_n, s}$;
- $p: s_1 \times \cdots \times s_n$ the fact that $p \in P_{s_1 \ldots s_n}$;
- $T_\Sigma(X)$ the *term algebra* on Σ and the S-sorted family of variables $X = \{X_s\}_{s \in S}$ and we write $t: s$ for $t \in (T_\Sigma(X))_s$;
- $\mathcal{FOF}_\Sigma(X)$ the set of the *first order formulas* (with possibly infinitary conjunctions) on Σ and X; if $\phi \in \mathcal{FOF}_\Sigma(X)$, then $fv(\phi)$ denotes the set of the *free variables* of ϕ.

A Σ-*algebra* A is a triple $(\{A_s\}_{s \in S}, \{f^A\}_{f \in F}, \{p^A\}_{p \in P})$ such that for all $s \in S$, A_s is a set, for all $f: s_1 \times \cdots \times s_n \to s$, $f^A: A_{s_1} \times \cdots \times A_{s_n} \to A_s$ is a total function and for all $p: s_1 \times \cdots \times s_n$, $p^A \subseteq A_{s_1} \times \cdots \times A_{s_n}$. If $\phi \in \mathcal{FOF}_\Sigma(X)$ and A is a Σ-algebra, we indicate as usual $A \models \phi$ the fact that ϕ holds in A.

A Σ-algebra is said *term generated* iff each element of a carrier is the interpretation of a ground term.

1.2 Some Examples

Here we give some examples adopting the well-known and accepted technique of viewing a process as a labelled transition system (see [M1]). A *labelled transition system* is a triple $TS = (S, L, \longrightarrow)$ where S is a set of *states*, L is a set of *labels* (or *flags*) and $\longrightarrow \subseteq S \times L \times S$ is the *transition relation*; as usual we write $s \xrightarrow{l} s'$ for $(s, l, s') \in \longrightarrow$.

Labelled transition systems can be seen as algebras on a signature having the sorts *state*, *label* and a predicate $\longrightarrow$: *state* $\times$ *label* $\times$ *state*; we call them *algebraic transition systems* (shortly, *ats*).

As a first example we rephrase the well-known (finite) *CCS* calculus of [M1] as an ats.

Example *CCS0*: **Finite** *CCS*

The signature of *CCS0* is the following, where we use the "–"-notation for defining mixfix operations:

$$
\begin{array}{l}
\textbf{sig}\ \Sigma_{CCS0} = \\
\quad \textbf{sorts}\ be, act \\
\quad \textbf{opns} \\
\qquad \textbf{nil}: \rightarrow be \\
\qquad - \cdot -: act \times be \rightarrow be \\
\qquad - + -: be \times be \rightarrow be \\
\qquad - \mid -: be \times be \rightarrow be \\
\qquad \{a: \rightarrow act \mid a \in ACT\} \\
\qquad \overline{-}: act \rightarrow act \\
\quad \textbf{preds} \\
\qquad - \xrightarrow{-} -: be \times act \times be
\end{array}
$$

where ACT is a set of operation symbols for actions such that $\tau \in ACT$.

The "usual" operational model for *CCS0* is just the *term-generated algebra* over the signature Σ_{CCS0} such that all and only the identifications which can be inferred from the equalities $\overline{\overline{a}} = a$ for all a: act, and $\overline{\tau} = \tau$ hold, and such that the interpretation of the predicate $\xrightarrow{-}$ is given by the following inductive rules (where a: act and b, b', b'', b_1, b_1': be):

$$a \cdot b \xrightarrow{a} b$$

$$\frac{b \xrightarrow{a} b'}{b + b'' \xrightarrow{a} b'} \qquad \frac{b \xrightarrow{a} b'}{b'' + b \xrightarrow{a} b'}$$

$$\frac{b \xrightarrow{a} b'}{b \mid b'' \xrightarrow{a} b' \mid b''} \qquad \frac{b \xrightarrow{a} b'}{b'' \mid b \xrightarrow{a} b'' \mid b'}$$

$$\frac{b \xrightarrow{a} b_1 \quad b' \xrightarrow{\bar{a}} b_1'}{b \mid b' \xrightarrow{\tau} b_1 \mid b_1'} \qquad \text{for } \alpha \neq \tau.$$

In the sequel we indicate this model simply by *CCS0*. **End of Example**

Thus in an ats processes are just a data type and so it is possible to formally describe systems where processere are exchanged as values, where there are functions taking as parameters and/or returning (values containing) processes and so on (see [AGR2]); as an example we give a simple variation of *CCS*.

Example CCS^+: **a Higher Order** *CCS*

We extend *CCS0* by allowing handshaking communication with exchange of behaviours (see [AR,T]); formally we add to the signature of *CCS0* an operation *SEND*: $be \rightarrow act$; a behaviour b can hence perform a $SEND(b')$ action, where b' is another behaviour, and the intuitive meaning is that b' is being sent as a value which can be received by some other process performing a corresponding $\overline{SEND(b')}$ action. **End of Example**

In this framework it is also possible to handle *concurrent systems*, i.e., a particular kind of transition systems in which a state has an internal structure built starting from another transitition system representing the (basic) active components of the state. As an example, here we build a concurrent system whose basic components are just *CCS*0 behaviours.

Example net-*CCS*: A net of *CCS* Behaviours
We add to the signature Σ_{CCS0}:

- a new sort ***net***, whose elements model nets of *CCS*0 behaviours (inductively defined as a single behaviour or a parallel composition $n_1||n_2$ of two networks) and whose activities proceed in a free parallel way, except when restricted by the "/" operation;
- a new sort ***lab***, whose elements are used to label the network transitions; network labels can be composed in parallel, and we assume to this end a binary operation on labels "$*$";
- a new transition relation $\Longrightarrow : net \times lab \times net$ on nets.

"net-*CCS*" is just the term-generated algebra over the enriched signature such that:

- all and only identifications on the new elements are due to the fact that: "$||$" and "$*$" are commutative and associative, τ is an identity for "$*$" and $\overline{a}$ is the inverse of a w.r.t. "$*$";
- the interpretation of the predicate $\Longrightarrow$ is given by means of the following inductive rules:

$$\frac{b \xrightarrow{a} b'}{b \overset{a}{\Longrightarrow} b'}$$

$$\frac{n_1 \overset{l}{\Longrightarrow} n_1'}{n_1||n_2 \overset{l}{\Longrightarrow} n_1'||n_2} \qquad \frac{n_2 \overset{l}{\Longrightarrow} n_2'}{n_1||n_2 \overset{l}{\Longrightarrow} n_1||n_2'}$$

$$\frac{n_1 \overset{l_1}{\Longrightarrow} n_1' \quad n_2 \overset{l_2}{\Longrightarrow} n_2'}{n_1||n_2 \overset{l_1*l_2}{\Longrightarrow} n_1'||n_2'} \qquad \frac{n \overset{l}{\Longrightarrow} n'}{n/l' \overset{l}{\Longrightarrow} n'/l'} \quad l \neq l'.$$

End of Example

2 Observational Structures

In sections 2.1 and 2.2 we motivate the formal definitions given in section 2.3 by means of the examples of section 1.

2.1 Similarity of experiments

Strong Bisimulation for *CCS0* Consider the ats *CCS0* given in section 1.2 formally defining *CCS*.

It is well-known that the above model is not satisfactory as a semantic model for *CCS*, since it distinguishes too much (for example, $b' + b''$ is different from $b'' + b'$); in this sense one is looking for better semantics for *CCS0*.

In general a semantics of an algebra A is given by means of a congruence on A; a congruence can be seen as an *A-family* satisfying additional constraints, where an *A-family* is couple $(\{R_s\}_{s\in S}, \{R_p\}_{p\in P})$, such that for all $s \in S$, $R_s \subseteq A_s^2$ and for all $p: s_1 \times \cdots \times s_n \in P$, $R_p \subseteq A_{s_1} \times \cdots \times A_{s_n}$. In particular, if R is a congruence on A, then A/R is the algebra modelling the semantics given by R (the *semantic model*).

Hence, in this framework, a semantics for *CCS0* is a couple $R = ((R_{act}, R_{be}), R_{\longrightarrow})$, where R_{act} and R_{be} are binary relations on $CCS0_{act}$ and $CCS0_{be}$ respectively, and $R_{\longrightarrow} \subseteq \longrightarrow^{CCS0}$.

The *strong bisimulation* semantics corresponds to the idea that two *CCS0* behaviours should be identified if and only if they behave in the same way if we can only observe the actions which label their transitions. As it is well known this semantics is given taking the quotient $CCS0/\sim$, where $\sim$ is the so-called *maximum strong bisimulation relation*.

A *CCS0*-family R is a *(strong) bisimulation relation* (see [P,M1]) iff

i) $b'\ R_{be}\ b''$ implies

- for all a: *act*, b_1': *be*, if $b' \xrightarrow{a} b_1'$ then there exists b_1'': *be* s.t. $b'' \xrightarrow{a} b_1''$ and $b_1'\ R_{be}\ b_1''$;
- for all a: *act*, b_1'': *be*, if $b'' \xrightarrow{a} b_1''$ then there exists b_1': *be* s.t. $b' \xrightarrow{a} b_1'$ and $b_1'\ R_{be}\ b_1''$;

ii) R_{act} is the identity relation;

iii) $R_{\longrightarrow} \subseteq \longrightarrow^{CCS0}$.

The *maximum strong bisimulation* $\sim$ does exist and is the union of all the strong bisimulations.

Now let us call $x \xrightarrow{a} b$, where x is a variable, an *experiment* for *CCS0*, for every a: *act* and every b: *be*; note that $x \xrightarrow{a} b$ is a first order formula, since $\longrightarrow$ is a predicate symbol. Then we can rephrase the definition of bisimulation replacing clause i) with the following:

i) $b'\ R_{be}\ b''$ implies

- for all experiments e' if b' *passes* e', then there exists a *similar* experiment e'', such that b'' *passes* e'' .
- for all experiments e'' if b'' *passes* e'', then there exists a *similar* experiment e', such that b' *passes* e'.

Clearly, if $e = x \xrightarrow{a} b$, "b' *passes* e" can be formally stated as "$e[b']$ holds in *CCS0*", where $e[b'] = e[b'/x] = b' \xrightarrow{a} b$, since $b' \xrightarrow{a} b$ is a first order formula. In this case we define $x \xrightarrow{a} b'$ to be *similar* to all and only the experiments of the form $x \xrightarrow{a} b''$ with

$b'\ R\ b''$. Notice that the similarity relation between experiments depends on R; hence we introduce a function C, that we call *similarity law*, associating with each R a binary relation $C(R)$ on experiments; in this case C is defined by: $x \xrightarrow{a} b'\ C(R)\ x \xrightarrow{a} b''$ iff $b'\ R\ b''$.

Weak Bisimulation If we decide that some actions, let us say τ actions, should not be observable, then we need a semantic equivalence which is less fine than strong bisimulation, since two behaviours whose activity differ only in the nonobservable actions performed should be made equivalent. This is achieved by defining the well-known *weak bisimulation*, which is obtained by introducing a new predicate $\Longrightarrow : be \times act \times be$ defined by the following inductive rules:

$$\frac{}{b \overset{\tau}{\Longrightarrow} b} \qquad \frac{b \xrightarrow{\tau} b' \quad b' \overset{a}{\Longrightarrow} b''}{b \overset{a}{\Longrightarrow} b''} \qquad \frac{b \overset{a}{\Longrightarrow} b' \quad b' \xrightarrow{\tau} b''}{b \overset{a}{\Longrightarrow} b''}.$$

This predicate introduces a different kind of experiments having form $x \overset{a}{\Longrightarrow} b$. Weak bisimulation is defined using the same definition schema of strong bisimulation by just changing the set of experiments and by using a similiarity relation analogous to the one used for strong bisimulation.

Divergence Sensitive Weak Bisimulation Let us extend *CCS*0 to include also some infinite behaviours (for example, either by means of a fixpoint combinator, or directly by means of recursive equations, as $\tau^\omega = \tau \cdot \tau^\omega$). It is well-known that weak bisimulation does not distinguish properly between terminating and nonterminating behaviours (for example, τ^ω is weakly equivalent to nil); to get a finer semantic equivalence we introduce a new kind of experiment, *Stop*, defined by the following infinitary first order formula:

$$Stop = \not\exists\{b_i, a_i\}_{i\in\omega}.(b_0 = x) \wedge (\bigwedge_{i\in\omega} b_i \xrightarrow{a_i} b_{i+1})$$

where the b_i's and a_i's are variables of sort *be* and *act* respectively. *Stop* succeeds on all and only the terminating behaviours. To be equivalent we require now that not only two behaviours have to exhibit the same visible actions, but they also have to agree w.r.t. termination. The definition schema of bisimulation rephrased using the concept of experiment handles already this case by taking as experiments $\{x \xrightarrow{a} b \mid a{:}\ act, b{:}\ be\} \cup \{Stop\}$ (and clearly *Stop* is only similar to itself), since clause i) is quantified on all experiments; the maximum bisimulation relation exists and identifies in this case all behaviours which behave similarly w.r.t. all of these experiments.

Observing Multilevel Parallelism It is useful to slightly generalize the definition schema by allowing several observed sorts, to be able to handle, for example, "net-*CCS*" (see section 1.2). In this case both the arrows $\rightarrow$ and $\Rightarrow$, representing the transitions of behaviours and of nets, can be used to build experiments for observing behaviours and nets, hence we have experiments of the form $x_{be} \xrightarrow{a} b$ and of the form $x_{net} \overset{l}{\Longrightarrow} n$; we want that the semantic identifications are made on behaviours and on nets accordingly to these

experiments. It is easy to extend the definition of bisimulation by quantifying clause i) over all observed sorts. Let $O = \{be, net\}$ be the set of *observed sorts*,

$$Exp = \{x_{be} \xrightarrow{a} b, x_{net} \stackrel{l}{\Longrightarrow} n \mid a\text{: } act, b\text{: } be, l\text{: } lab, n\text{: } net\}$$

the set of *experiments*, and for all R let $\mathcal{C}(R)$ be the following similarity relation:

$$x_{be} \xrightarrow{a} b' \; \mathcal{C}(R) \; x_{be} \xrightarrow{a} b'' \quad \text{iff} \quad b' \; R_{be} \; b''$$

and

$$x_{net} \stackrel{l}{\Longrightarrow} n' \; \mathcal{C}(R) \; x_{net} \stackrel{l}{\Longrightarrow} n'' \quad \text{iff} \quad n' \; R_{net} \; n''.$$

A net-*CCS*-family is a *multilevel bisimulation* iff

i) for all $o \in O$, $t' \; R_o \; t''$ implies for all $e' \in Exp$ with free variable of sort o

- if $e'[t']$ holds, then there exists $e'' \in Exp$ such that $e''[t'']$ holds and $e' \; \mathcal{C}(R) \; e''$;
- if $e'[t'']$ holds, then there exists $e'' \in Exp$ such that $e''[t']$ holds and $e'' \; \mathcal{C}(R) \; e'$;

ii) for all $s \notin O$ R_s is the identity relation;

iii) for all $p \in P$, $R_p \subseteq p^A$.

Since $\mathcal{C}$ is monotonic, then there exists the maximum multilevel bisimulation, which is also the maximum fixed point of an appropriate function.

2.2 Propagating Identities

In the examples introduced in the previous section, the semantics of the objects of the nonobserved sorts *act* and *lab* is fixed: the semantic identifications made on behaviours (and on nets) do not introduce new identifications on actions and labels. Clearly, this is not always the case, and we explain this point by considering the case of CCS^+ (see section 1.2).

In this case we want that, given b' and b'', if b' is semantically equivalent to b'' then also the action $SEND(b')$ should be semantically equivalent to $SEND(b'')$. The propagation of the semantic identifications to other sorts is represented by means of a *propagation function* $\mathcal{P}$, for all $s \in S$, $\mathcal{P}(R)_s$ is the propagation of R to the elements of sort s (we require $\mathcal{P}(R)_o = R_o$ for all $o \in O$). In this case we have that given R, if $b' \; R \; b''$, then $SEND(b') \; \mathcal{P}(R) \; SEND(b'')$, so the propagation law $\mathcal{P}$ is defined for all R as follows:

$$\begin{aligned} \mathcal{P}(R)_{act} \;=\; & \{(a,a), (\overline{a},\overline{a}) \mid a \in ACT\} \cup \\ & \{(SEND(b'), SEND(b'')), (\overline{SEND(b')}, \overline{SEND(b'')}) \mid b' \; R_{be} \; b''\}. \end{aligned}$$

To complete the example, we have to define the similarity relation between experiments: it seems reasonable to consider a generic experiment $x \xrightarrow{a} b$ to be equivalent

to all the experiments of the form $x \xrightarrow{a'} b'$ with $a\ \mathcal{P}(R)\ a'$ and $b\ R\ b'$. In particular if $a = SEND(b_1)$, then

$$x \xrightarrow{SEND(b_1)} b \quad \text{is similar to} \quad x \xrightarrow{SEND(b_2)} b'$$

for all $b\ R\ b'$, $b_1\ R\ b_2$. Hence the similarity law $\mathcal{C}$ can be defined in this case in terms of $\mathcal{P}$ as follows: for all R

$$x \xrightarrow{a} b\ \mathcal{C}(R)\ x \xrightarrow{a'} b'$$

for all a, a', b, b' such that $a\ \mathcal{P}(R)\ a'$, $b\ R\ b'$.

2.3 Observational Structures and their Semantics

The discussions, definitions and examples of the previous sections are collected in the notion of *observational structure* and of *(maximum) observational relation*, which are a general framework for observational semantics which is not only restricted to concurrency (see [AGR1]), even though all the applications shown in this paper are to concurrency.

In this section A denotes a Σ-algebra on a signature $\Sigma = (S, F, P)$, and $O \subseteq S$ denotes the set of the observed sorts. A semantics on A is represented by an *A-family* which is defined as follows.

Def. 2.1 *For $S' \subseteq S$, an* (A, S')-family *is an S'-indexed family $R = \{R_s\}_{s \in S'}$ s.t. for all $s \in S'$ $R_s \subseteq A_s^2$.*

A couple $(R_S, \{R_p\}_{p \in P})$, where R_S is an (A, S)-family and $R_p \subseteq A_{s_1} \times \cdots \times A_{s_n}$ for all $p: s_1 \times \cdots \times s_n \in P$, is called A-family.

If R is an A-family and $S' \subseteq S$, then $R|_{S'}$ indicates the (A, S')-family $\{R_s\}_{s \in S'}$.

A family R is reflexive *iff for all s R_s is reflexive; similarly for* symmetric, transitive *and* an equivalence. □

Def. 2.2 *The set of* experiments *in Σ on O, indicated with* $\mathbf{Exp}(\Sigma, O)$, *is defined as follows:*

$$\mathbf{Exp}(\Sigma, O) = \{\phi \in \mathcal{FOF}_\Sigma(X) \mid card(fv(\phi)) = 1 \wedge fv(\phi) \subseteq \bigcup_{o \in O} \{x_o\}\}.$$

If $fv(e) = \{x_o\}$ we write $e: o$. □

Given an experiment $e \in \mathbf{Exp}(\Sigma, O)$ such that $e: o$, an element $a \in A_o$ and a valuation v s.t. $v(x_o) = a$, we write $A \models e[a]$ to indicate that e holds in A under the valuation v. Usually we do not insist in specifying the sort of an experiment whenever this is clear from the context.

Def. 2.3 (Similarity Laws)
S-law(A, O) *indicates the set of all monotonic functions from A-families into the set of binary relations on* $\mathbf{Exp}(\Sigma, O)$ *respecting the sorts of the experiments.* □

Def. 2.4 (Propagation Laws)
P-law(A,O) *indicates the set of all monotonic functions* $\mathcal{P}$ *from* (A,O)*-families into* A*-families s.t.* $\mathcal{P}(R)_o = R_o$ *for all* $o \in O$. □

The fact that similarity and propagation laws are monotonic is needed to prove prop. 2.8.
In section 3.4 we use the notation $\mathcal{P}_A$ to indicate the propagation law s.t.:

- $\mathcal{P}_A(R)_s = \{(a,a) \mid a \in A_s\}$ for all $s \in S - O$;
- $\mathcal{P}_A(R)_p = p^A$ for all $p \in P$.

Def. 2.5 (Observational Structures)
An observational structure *is a* 6*-uple* $(\Sigma, A, O, Exp, \mathcal{C}, \mathcal{P})$ *where*

- $\Sigma = (S, F, P)$ *is a signature;*
- A *is a* Σ*-algebra (the structure on which we want to define a semantics);*
- $O \subseteq S$ *is a set of sorts (*observed sorts*, the sorts of the objects on which we perform some experiments);*
- $Exp \subseteq \mathbf{Exp}(\Sigma, O)$;
- $\mathcal{C} \in \mathbf{S\text{-}law}(A, O)$;
- $\mathcal{P} \in \mathbf{P\text{-}law}(A, O)$. □

In the following we use OS to indicate a generic observational structure $(\Sigma, A, O, Exp, \mathcal{C}, \mathcal{P})$.

Def. 2.6 *An* A*-family* R *is an* observational relation *for* OS *(shortly, an* o-relation*) iff*

i) $\forall o \in O$, $\forall a', a'' \in A_o$ $a'\ R_o\ a''$ *implies*
  - $*$ $\forall e' \in Exp$, $A \models e'[a']$ *implies* $\exists e'' \in Exp$ *s.t.* $e'\ \mathcal{C}(R)\ e''$ *and* $A \models e''[a'']$;
  - $**$ $\forall e'' \in Exp$, $A \models e''[a'']$ *implies* $\exists e' \in Exp$ *s.t.* $e'\ \mathcal{C}(R)\ e''$ *and* $A \models e'[a']$;

ii) $\forall s \in S - O$, $R_s \subseteq \mathcal{P}(R\mid_O)_s$;

iii) $\forall p \in P$, $R_p \subseteq \mathcal{P}(R\mid_O)_p$. □

As for the case of strong bisimulation, for each OS there is a monotonic function $\mathcal{F}_{OS}$ on A-families, which can be used to characterize the observational relations and whose maximum fixed point (which does always exist) is the maximum observational relation.

Def. 2.7 *For all* A*-families* R,

$$\mathcal{F}_{OS}(R) = \mathcal{P}(\{\{(a', a'') \mid a', a'' \in A_o,\ * \text{ and } ** \text{ hold}\}\}_{o \in O}).$$ □

Prop. 2.8 *The following facts hold:*

1. *an* A*-family* R *is an o-relation iff* $R \subseteq \mathcal{F}_{OS}(R)$;
2. $\mathcal{F}_{OS}$ *is monotonic over the complete lattice of* A*-families, ordered by inclusion;*

3. the (arbitrary) union of o-relations is an o-relation;

4. $\sim_{OS} =_{\text{def}} \bigcup\{R | R \subseteq \mathcal{F}_{OS}(R)\}$ *is an o-relation and* $\sim_{OS} = \text{maxfix}\, \mathcal{F}_{OS}$. □

Sometimes we indicate $\sim_{OS}$ simply by $\sim$ and call it *the maximum* o-relation of OS.

Notice that $a' \sim a''$ iff there exists an o-relation R s.t. $a'\ R\ a''$; moreover $(a_1, \ldots, a_n) \in \sim_p$ iff there exists an o-relation R s.t $(a_1, \ldots, a_n) \in R_p$.

In general we cannot ensure the maximum o-relation to be either reflexive, or transitive, or symmetric; to this end additional requirements on $\mathcal{P}$ and $\mathcal{C}$ can be made; we show just an example.

Prop. 2.9 *If for all A-families R ve have that* $\mathcal{C}(R^*) = \mathcal{C}(R)^*$ *and if for all equivalences R we have that* $\mathcal{P}(R)$ *is an equivalence, then* $\sim$ *is an equivalence, where* R^* *indicates the smallest equivalence containing R.* □

If $\mathcal{C}$ and $\mathcal{P}$ are as in prop. 2.9, then we say that $\mathcal{C}$ *reflects equivalences* and $\mathcal{P}$ *propagates equivalences.*

Even when $\sim$ is an equivalence, it may be that it is not a congruence (for example, the case of weak bisimulation). Sufficient conditions ensuring $\sim$ to be a congruence can be found for the case of transition systems in [GV] and for the algebraic case in [GR].

In the cases when $\sim$ is not a congruence, one can also proceed in a similar way to what has been done by Milner in [M2] for the case of weak bisimulation, and take the greatest congruence contained in $\sim$; in our framework this corresponds to replacing each experiment $e: o$ by the set of experiments $e[c[x_{o'}]]$ for all contexts $c[x_{o'}]: o$, for all $o' \in O$.

Example The observational structure implicitely used in section 1.2 to define strong bisimulation semantics for $CCS0$ is

$$(CCS0, be, \{x \xrightarrow{a} b \mid a: act, b: be\}, \mathcal{C}_{CCS0}, \mathcal{P}_{CCS0})$$

where $x \xrightarrow{a} b'\ \mathcal{C}_{CCS0}(R)\ x \xrightarrow{a} b''$ iff $b'\ R\ b''$.

End of Example

2.3.1 Testing Structures

Testing structures are a very simple but important class of observational structures used in section 3 to state and prove the generalized version of Hennessy-Milner theorem. They generalize the framework of testing semantics for processes introduced in [DH] and are essentially observational structures where two experiments are similar iff they are the same experiment.

Def. 2.10 *A testing structure is an observational structure* $(\Sigma, A, O, Exp, \mathcal{ID}, \mathcal{P})$, *where* $\mathcal{ID}$ *is the similarity law defined by* $\mathcal{ID}(R) = \{(e', e'') \mid e', e'' \in Exp$ *logically equivalent*$\}$, *for all R.* □