The total order assumption

5

The Total Order Assumption

J.C.M. Baeten
Department of Computing Science, Eindhoven University of Technology,
P.O.Box 513, 5600 MB Eindhoven, The Netherlands

The total order assumption (TOA) is the assumption that all execution sequences of observable actions or events are totally ordered by precedence. As long as in some cases TOA is to the point, total order and partial order semantics are both legitimate but lead to different theories. I argue that TOA is a simplifying assumption, and present an example of a total order theory without interleaving, and a partial order theory with interleaving.

1980 Mathematics Subject Classification (1985 revision): 68Q55, 68Q45, 68Q10.
1987 CR Categories: F.1.2, F.3.1, D.1.3, D.3.1.
Key words & Phrases: partial order, total order, true concurrency, interleaving, process algebra.
Note: This research was carried out in the context of ESPRIT Basic Research Action 7166, CONCUR2.

1. INTRODUCTION.

Semantics of concurrency like Petri nets [REI85] or event structures [WIN87] are often called *True Concurrency* semantics, in order to reflect the viewpoint that such semantics embody a truer treatment of concurrency than interleaving semantics like the standard semantics of CCS [MIL89], CSP [HOA85] or ACP [BAW90]. Another name for True Concurrency semantics is *partial order* semantics. In partial order semantics, execution paths of observable actions or events are partially ordered by precedence.

Semantics that do not classify as partial order semantics (like CCS, CSP, ACP semantics) are sometimes jokingly called False Concurrency semantics, but are usually referred to as *interleaving* semantics. An interleaving semantics is a semantics in which parallel composition of finite processes can be expressed in terms of some form of alternative composition and some form of sequencing. An interleaving semantics will contain some form of the *Expansion Theorem*, that can be used to remove parallel composition from a finite process expression, and can be used to unfold the parallel composition of recursively defined processes.

In this article, I will point out that the division of semantics into partial order semantics and interleaving semantics does not give a partition of the field. In order to get a clearer division, I propose to use the name *total order* semantics. A total order semantics is a semantics that satisfies the *Total Order Assumption (TOA)*. This assumption states that all execution sequences of observable actions or events are totally ordered by precedence. As long as in some cases TOA is to the point, total order and partial order semantics are both legitimate (but lead to different theories). I will argue that TOA is a simplifying assumption, and present an example of a total order theory without interleaving, and a partial order theory with interleaving.

ACKNOWLEDGEMENTS.
I thank J.A. Bergstra (University of Amsterdam) for extensively discussing the contents of this paper with me. At the workshop "What good is partial order" on 22 June 1992 at Sheffield, I discussed this paper with E. Best (University of Hildesheim), R. Milner (University of Edinburgh) and others. Further input came from S. Mauw (Eindhoven University of Technology).

2. THE TOTAL ORDER ASSUMPTION.

The division of semantics into partial order semantics like Petri nets [REI85] or event structures [WIN87], and interleaving semantics like the standard semantics of CCS [MIL80], CSP [HOA85] or ACP [BAW90] is not an exhaustive division. Moreover, the two notions overlap. In order to get a clearer division, I propose to use the name *total order* semantics. A total order semantics is a semantics that satisfies the *Total Order Assumption (TOA).*

THE TOTAL ORDER ASSUMPTION (TOA):
All execution sequences of observable actions or events are totally ordered by precedence.

In contrast, in partial order semantics the execution sequences or execution paths need not be totally ordered, but are only partially ordered (by causality). We see that total order semantics form a special case of partial order semantics, just like, in mathematics, the theory of ordinary differential equations is a special case of the theory of partial differential equations, or linear mathematics is a special case of general mathematics. This subclass relation gives a better picture of the relation between the two types of semantics than opposites with emotional meaning like True vs. False, or Right vs. Wrong.

I want to emphasise the difference between interleaving semantics and total order semantics. I will do this by presenting, in section 4, a total order semantics without an expansion theorem, and presenting, in section 5, a partial order semantics with an expansion theorem.

There is sometimes a debate going on between advocates of partial order semantics and total order semantics as to which kind is better. I think this kind of debate is pointless. One does not ask if the theory of partial differential equations is better than the theory of ordinary differential equations. As long as both theories can be usefully applied in some cases, they are both legitimate (but have their own characteristics). The same holds for semantic theories, so we have the following thesis.

As long as in *some* cases TOA is to the point,
total order and partial order semantics are both legitimate.

Thus, we have to avoid all claims of superiority when we talk about our favorite theory. I will never say that ACP preaches the TOA, or that ACP justifies the TOA, or even that ACP prefers the TOA. The only relevant question that we can ask about a semantics in one of the two categories is, whether or not it exploits the basic tenets of that class in a best possible way, whether or not it takes full advantage of the characteristics of that class. Thus, a relevant question is: does ACP exploit TOA in a best possible way? I think it does, and will give some motivation in the following.

A somewhat controversial thesis is the following.

TOA is a simplifying assumption.

Some people may argue that this thesis is not correct, as some methods of analysis are perhaps easier to apply in a more general framework. Specifically, methods for model checking in interleaving semantics usually consider the whole state space of a process, and the size of the state space can be very large compared to the size of the process expression. This blow-up in size is known as *combinatorial explosion*. In partial order semantics, we cannot calculate the whole state space, and easier methods may be found (see [ESP92]). Of course, we have to realise that since total order semantics is a special case of partial order semantics, any method found in the wider framework can be transfered to the more restricted framework.

The main argument in favor of the simplifying nature of the TOA is the *calculational* aspect: using the TOA, it is easier to calculate with process expressions, and it is easier to give correctness proofs in a linear form, amenable to automation. Such a calculus is often based on a set of axioms or laws. We give an example of an algebraic correctness proof in the following section 3. I claim that a proof like this calculation cannot be given in partial order semantics (at least, I have never seen one, nor have I seen a usable complete axiomatisation of a partial order semantics).

Another argument in favor of the simplifying nature of the TOA is extensibility: it is easier to extend a theory based on total order semantics to incorporate new features. As an example, we consider real time extensions in section 4. It is more difficult to achieve such extensions in the full generality of partial order semantics.

3. EXAMPLE: THE ALTERNATING BIT PROTOCOL.

The standard example in concurrency theory is the Alternating Bit Protocol. In this section, we give a purely calculational (in fact algebraic) proof of the correctness of this protocol. Note that no pictures or other visual aids are used in the proof. Essentially, we give the proof of [BEK86]. The specification is as follows. B is the set of Booleans, D is a finite data set, and we use the standard send/receive communication function of [BAW90] that only allows communications of the form $s_i(x) \mid r_i(x) = c_i(x)$ (i a port name).

Sender:

$$S = S0 \cdot S1 \cdot S$$

$$Sb = \sum_{d \in D} r_1(d) \cdot Sb_d \qquad b \in B$$

$$Sb_d = s_2(db) \cdot Tb_d \qquad b \in B, d \in D$$

$$Tb_d = (r_6(1\text{-}b) + r_6(error)) \cdot Sb_d + r_6(b) \qquad b \in B, d \in D.$$

Channels:

$$K = \sum_{f\in D\times B} r_2(f)\cdot(i\cdot s_3(f) + i\cdot s_3(\text{error}))\cdot K \ .$$

$$L = \sum_{b\in B} r_5(b)\cdot(i\cdot s_6(b) + i\cdot s_6(\text{error}))\cdot L.$$

Receiver:

$$R = R1\cdot R0\cdot R$$

$$Rb = (r_3(\text{error}) + \sum_{d\in D} r_3(db))\cdot s5(b)\cdot Rb + \sum_{d\in D} r_3(d(1\text{-}b))\cdot s_4(d)\cdot s_5(1\text{-}b) \qquad b\in B.$$

Encapsulation:

$$H = \{s_p(x), r_p(x) : p \in \{2,3,5,6\}, x \in D\times B \cup B \cup \{\text{error}\}\}.$$

Abstraction:

$$I = \{c_p(x) : p \in \{2,3,5,6\}, x \in D\times B \cup B \cup \{\text{error}\}\} \cup \{i\}.$$

Specification of the protocol:

$$ABP = \tau_I \circ \partial_H(S \parallel K \parallel L \parallel R).$$

We want to prove: $ABP = \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot ABP$.

We give the complete calculation that proves this fact. First, we derive a guarded recursive specification for $i_I \circ \partial_H(S \parallel K \parallel L \parallel R)$. The operator i_I is pre-abstraction (i.e. all internal actions are renamed into i, but no τ-laws can be applied). We will use the following abbreviations:

$X = i_I\circ\partial_H(S \parallel K \parallel L \parallel R)$ $\qquad$ $Y = i_I\circ\partial_H(S1\cdot S \parallel K \parallel L \parallel R0\cdot R)$

$X1_d = i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R)$ $\qquad$ $Y1_d = i_I\circ\partial_H(S1_d\cdot S \parallel K \parallel L \parallel R0\cdot R)$

$X2_d = i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(0)\cdot R0\cdot R)$ $\qquad$ $Y2_d = i_I\circ\partial_H(T1_d\cdot S \parallel K \parallel L \parallel s_5(1)\cdot R)$

$K'_f = (i\cdot s_3(f) + i\cdot s_3(\text{error}))\cdot K$ $\qquad$ $L'_b = (i\cdot s_6(b) + i\cdot s_6(\text{error}))\cdot L.$

Each step in the following calculation is an application of the expansion theorem.

$$X = i_I\circ\partial_H(S \parallel K \parallel L \parallel R) = \sum_{d\in D} r_1(d)\cdot i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R) = \sum_{d\in D} r_1(d)\cdot X1_d$$

$$\begin{aligned}
X1_d &= i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R) = i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K'_{d0} \parallel L \parallel R) = \\
&= i\cdot(i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel s_3(d0)\cdot K \parallel L \parallel R) + i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel s_3(\text{error})\cdot K \parallel L \parallel R)) = \\
&= i\cdot(i^2\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_4(d)\cdot s_5(0)\cdot R0\cdot R) + \\
&\qquad\qquad + i^2\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(1)\cdot R1\cdot R0\cdot R)) = \\
&= i\cdot(i^2\cdot s_4(d)\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(0)\cdot R0\cdot R) + i^3\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L'_1 \parallel R)) = \\
&= i\cdot(i^2\cdot s_4(d)\cdot X2_d + i^3\cdot(i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel s_6(1)\cdot L \parallel R) + \\
&\qquad\qquad\qquad\qquad + i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel s_6(\text{error})\cdot L \parallel R))) = \\
&= i\cdot(i^2\cdot s_4(d)\cdot X2_d + i^3\cdot(i^2\cdot i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R) + i^2\cdot i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R))) = \\
&= i\cdot(i^2\cdot s_4(d)\cdot X2_d + i^5\cdot X1_d)
\end{aligned}$$

$$\begin{aligned}
X2_d &= i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(0)\cdot R0\cdot R) = i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L'_0 \parallel R0\cdot R) = \\
&= i\cdot(i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel s_6(0)\cdot L \parallel R) + i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel s_6(\text{error})\cdot L \parallel R)) = \\
&= i\cdot(i^2\cdot i_I\circ\partial_H(S1\cdot S \parallel K \parallel L \parallel R0\cdot R) + i^2\cdot i_I\circ\partial_H(S0_d\cdot S1\cdot S \parallel K \parallel L \parallel R0\cdot R)) = \\
&= i\cdot(i^2\cdot Y + i^3\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K'_{do} \parallel L \parallel R0\cdot R)) =
\end{aligned}$$

$$= i\cdot(i^2\cdot Y + i^3\cdot(i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel s_3(d0)\cdot K \parallel L \parallel R0\cdot R) +$$
$$+ i\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel s_3(error)\cdot K \parallel L \parallel R0\cdot R))) =$$
$$= i\cdot(i^2\cdot Y + i^3\cdot(i^2\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(0)\cdot R0\cdot R) +$$
$$+ i^2\cdot i_I\circ\partial_H(T0_d\cdot S1\cdot S \parallel K \parallel L \parallel s_5(0)\cdot R0\cdot R))) =$$
$$= i\cdot(i^2\cdot Y + i^5\cdot X2_d).$$

Likewise, we derive

$$Y = \sum_{d\in D} r_1(d)\cdot Y1_d$$
$$Y1_d = i\cdot(i^2\cdot s_4(d)\cdot Y2_d + i^5\cdot Y1_d)$$
$$Y2_d = i\cdot(i^2\cdot X + i^5\cdot Y2_d).$$

The equations derived for $X1_d$ and $X2_d$ show an internal loop of six steps. Application of Koomen's Fair Abstraction Rule ($KFAR_6$, see [BEK86] or [BAW90]) yields:

$$\tau\cdot\tau_I(X1_d) = \tau\cdot s_4(d)\cdot\tau_I(X2_d) \qquad \tau\cdot\tau_I(X2_d) = \tau\cdot\tau_I(Y)$$

Using this,

$$\tau_I(X) = \sum_{d\in D} r_1(d)\cdot\tau_I(X1_d) = \sum_{d\in D} r_1(d)\cdot\tau\cdot\tau_I(X1_d) = \sum_{d\in D} r_1(d)\cdot\tau\cdot s_4(d)\cdot\tau_I(X2_d) =$$
$$= \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot\tau\cdot\tau_I(X2_d) = \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot\tau\cdot\tau_I(Y),$$

so $$\tau_I(X) = \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot\tau_I(Y).$$

Similarly $$\tau_I(Y) = \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot\tau_I(X).$$

Application of the Recursive Specification Principle (RSP, see [BEK86] or [BAW90]) yields $\tau_I(X) = \tau_I(Y)$, and so we have obtained

$$ABP = \sum_{d\in D} r_1(d)\cdot s_4(d)\cdot ABP\ .$$

I claim that a proof of this nature cannot be given in partial order semantics. Further, I claim that proofs of this kind are very amenable to support by software tools, that can handle most of the calculations, and can incorporate many proof heuristics.

4. REAL TIME PROCESS ALGEBRA.

First, we take a look at ACPρ, the extension of ACP with real time aspects described in [BAB91a]. In total order semantics, an action or event is supposed to be observed at a certain moment in time. That is why such actions are called *atomic* and have no duration. If we want to talk about time explicitly, it makes sense therefore to attach the moment of observation to the atomic action. Here, we use absolute, global time.

So, if a is an atomic action and t a moment in time ($t \in \mathbb{R}_{\geq 0}$), then a(t) means that a takes place (is observed) at time t. ACPρ starts from these timed actions, and has the same operators as ACP. To give an example, we have

$$(a(2)\cdot c(4)) \parallel b(3) = a(2)\cdot b(3)\cdot c(4),$$

and we see that the execution sequence is totally ordered by time. If we use a totally ordered set like the non-negative reals for our time domain, it is natural to use a total order semantics.

ACPρ has interleaving axioms and an expansion theorem, as might be expected of a total order theory. We give some typical axioms:

$x \parallel y = x \mathbin{|\!\!\lfloor} y + y \mathbin{|\!\!\lfloor} x + x \mid y$

$a(t) \mathbin{|\!\!\lfloor} x = a(t)\cdot x$ if $t < U(x)$ (i.e., if x can wait until t)

$a(t) \mathbin{|\!\!\lfloor} x = \delta(U(x))$ (i.e., $a(t) \mathbin{|\!\!\lfloor} x$ deadlocks at time $U(x)$)
if $t \geq U(x)$ (i.e., if x cannot wait until t)

$a(t)\cdot x \mathbin{|\!\!\lfloor} y = a(t)\cdot(x \parallel y)$ if $t < U(y)$

$a(t)\cdot x \mathbin{|\!\!\lfloor} y = \delta(U(y))$ if $t \geq U(y)$.

An interesting feature of ACPρ (essential to describe many examples) is the presence of *integration*. The term $\int_{t\in(1,2)} a(t)$ represents the process that can execute action a somewhere between time 1 and time 2.

The theory with integration also has an expansion theorem. If we limit ourselves to *prefixed integration* (i.e. only integration over intervals, and the operand starts with an action containing the time variable), then we can still obtain a complete axiomatisation. As a result, we obtain that equality of finite processes is decidable (see [KLU91], [FOK92]).

Other real time process algebras do not have a concept of integration with explicit time variables like ACPρ. An example of this is the theory TCCS introduced by [MOT89]. Variants of TCCS are given in [MOT90] and [YI90]. An interesting result about the dense time variant of TCCS was recently given in [GOL92]. Consider the TCCS term $\varepsilon(1)\cdot a \mid b$. In our notation, this would be the process expression

$$\int_{t\geq 1} a(t) \parallel \int_{v\geq 0} b(v).$$

By the expansion theorem of ACPρ, this expression can be reduced to

$$\int_{t\geq 1} a(t)\cdot \int_{v\geq 0} b(v) + \int_{v\geq 0} b(v)\cdot \int_{t\geq \max(1,v)} a(t).$$

[GOL92] shows that the term in TCCS cannot be reduced to an expression without parallel composition, and they show in general that there are TCCS-terms with n parallel components that cannot be written with fewer parallel components. Thus, there is no expansion theorem for TCCS, and TCCS has a total order semantics (because its semantics is based on a form of bisimulation) that is not an interleaving semantics. This result probably holds in general for all dense time process algebras without explicit time variables such as versions of ATP [NSY91].

The presence of an expansion theorem is a key feature for a total order theory, that allows calculations used in correctness proofs like the proof of the ABP above. The absence of an expansion theorem means therefore that the TOA is not optimally exploited, and we can draw the following conclusion.

TCCS exploits the TOA to a lesser extent than ACPρ.

Another interesting extension of a process algebra is the extension to probabilistic choice. An axiomatic approach to ACP with probabilistic choice, featuring an expansion theorem, is given in [BABS92]. I claim that the introduction of new features like time or probabilities is so much more complex in partial order semantics, that the key issues and main difficulties do not stand out so easily.

5. FROM TOA TO POA.

The theory ACPρ presented in the previous section can be further extended to take place coordinates into account [BAB91b]. Then, atomic actions are parametrized by four reals denoting a point in space and time.

First of all, we can consider this using classical, Newtonian space/time. We obtain a total order theory, with interleaving and an expansion theorem, of which all laws are invariant under Galilei transformations (classical change of coordinates). An example, with $x_0, x_1 \in \Re^3$:

$$a(x_0, 1) \cdot b(x_0, 2) \cdot c(x_1, 3) \parallel d(x_0, 2) \cdot e(x_0, 3) \cdot f(x_1, 5) =$$

$$= a(x_0, 1) \cdot (b \mid d)(x_0, 2) \cdot (e(x_0) \,\&\, c(x_1))(3) \cdot f(x_1, 5).$$

Here, $b \mid d$ denotes *communication*, actions taking place at the same place and time, and $e(x_0) \,\&\, c(x_1)$ denotes *synchronisation*, actions taking place at the same time, but at a different place. The last construct is called a *multi-action*. This theory was used to describe communication between moving objects (e.g. satellite communication) in [BAB91c].

The situation changes if we consider relativistic, Einsteinian space/time. Then, we do not have a total ordering on events any longer, but only a partial ordering:

for $\alpha, \beta \in \Re^4$: $\alpha < \beta$ iff for *all* observers the time of α is before the time of β. Consider fig. 1 (note that we start using pictures at the moment we turn to partial order semantics). The vertical axis is the time axis. The two other axes suggested by the drawing represent the three spatial dimensions.

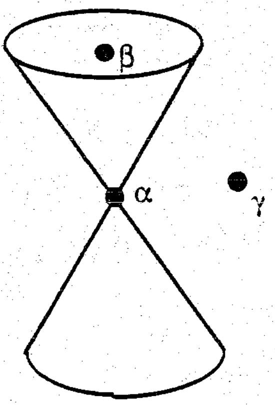

FIGURE 1.

In figure 1, we have $\alpha < \beta$ and we say β is inside the *positive light cone* of α (i.e. it is possible to travel from α to β with a speed less than the speed of light), whereas α and γ are incomparable, $\alpha \# \gamma$.

Now we can still use interleaving axioms as in the classical case, and obtain a total order theory with interleaving. To give an example, if $\alpha \# \gamma$ then we obtain $a(\alpha) \parallel c(\gamma) = a(\alpha) \cdot \delta + c(\gamma) \cdot \delta$, since, *after* executing a, c cannot be executed any more. This theory interleaves all actions on a single processor, we can call this *temporal interleaving* (see [BAB91b]). If we want a and c to execute independently in the example above, we cannot use the interleaving axioms as above, and we go to a partial order theory. This is the *multiple processor space/time process algebra* of [BAB91b]. The surprising observation of this paper is, that we can still formulate a formal expansion theorem, thus obtaining a partial order theory with an expansion theorem.

To this end, we add a new sequencing operator $\circ$, and $a(\alpha)\circ x$ now only excludes that part of x that precedes α, that is in the negative light cone of α (whereas $a(\alpha)\cdot x$ excludes that part of x that is not in the positive light cone of α, only allows points after α). If $\alpha \# \gamma$ then we obtain

$$a(\alpha) \parallel c(\gamma) = a(\alpha) \mathbin{\|\mkern-9mu\lfloor} c(\gamma) + c(\gamma) \mathbin{\|\mkern-9mu\lfloor} a(\alpha) = a(\alpha)\circ c(\gamma) + c(\gamma)\circ a(\alpha).$$

For any observer, this system will show only one behaviour, and not a choice of two behaviours. The sum here does not denote a choice between alternatives, but two different possibilities of observation. We have here a formal expansion with no clear operational intuition. Still, we have a partial order theory, with interleaving and an expansion theorem, of which all laws are Lorentz invariant (remain valid after a Lorentz transformation, a relativistic change of coordinates).

6. CONCLUDING REMARKS.

1. Total order semantics and partial order semantics are both legitimate (but lead to different theories).
2. In some cases, the Total Order Assumption is appropriate and useful.
3. The Total Order Assumption is a simplifying assumption.
4. Total order style mathematics (ACP, CCS, CSP, Floyd-Hoare logic, etc.) is an adequate exploitation of the Total Order Assumption.
5. Dense time process algebra requires explicit time variables for an optimal exploitation of the Total Order Assumption.
6. Partial order style mathematics (Petri nets, event structures, etc.) is very important in all cases where the TOA is not appropriate or to the point.

REFERENCES.

[BAB91a] J.C.M. BAETEN & J.A. BERGSTRA, *Real time process algebra*, Formal Aspects of Computing 3 (2), 1991, pp. 142-188.

[BAB91b] J.C.M. BAETEN & J.A. BERGSTRA, *Real space process algebra*, in: Proc. CONCUR'91, Amsterdam (J.C.M. Baeten & J.F. Groote, eds.), Springer LNCS 527, 1991, pp. 96-110.

[BAB91c] J.C.M. BAETEN & J.A. BERGSTRA, *Asynchronous communication in real space process algebra*, in: Proc. Formal Techniques in Real-Time and Fault-Tolerant Systems, Nijmegen 1992 (J. Vytopil, ed.), Springer LNCS 571, 1991, pp. 473-492.

[BABS92] J.C.M. BAETEN, J.A. BERGSTRA & S.A. SMOLKA, *Axiomatizing probabilistic processes: ACP with generative probabilities*, to appear in Proc. CONCUR'92, Stony Brook 1992.

[BAW90] J.C.M. BAETEN & W.P. WEIJLAND, *Process algebra*, Cambridge Tracts in Theor. Comp. Sci. 18, Cambridge University Press 1990.

[BEK86] J.A. BERGSTRA & J.W. KLOP, *Verification of an alternating bit protocol by means of process algebra*, in: Math. Methods of Spec. and Synthesis of Software Systems '85 (W. Bibel & K.P. Jantke, eds.), Springer LNCS 215, 1986, pp. 9-23.

[ESP92] J. ESPARZA, *Fast model checking using partial order semantics*, Workshop "What good is partial order", Sheffield 1992.

[FOK92] W.J. FOKKINK & A.S. KLUSENER, *Real time process algebra with prefixed integration*, report CS-R92xx, CWI Amsterdam 1992.

[GOL92] J.C. GODSKESEN & K.G. LARSEN, *Real-time calculi and expansion theorems*, report, Aalborg University 1992.

[HOA85] C.A.R. HOARE, *Communicating sequential processes*, Prentice Hall 1985.

[KLU91] A.S. KLUSENER, *Completeness in real time process algebra*, in: Proc. CONCUR'91, Amsterdam (J.C.M. Baeten & J.F. Groote, eds.), Springer LNCS 527, 1991, pp. 376-392.

[MIL89] R. MILNER, *Communication and concurrency*, Prentice Hall 1989.

[MOT89] F. MOLLER & C. TOFTS, *A temporal calculus of communicating systems*, report LFCS-89-104, University of Edinburgh 1989.

[MOT90] F. MOLLER & C. TOFTS, *A temporal calculus of communicating systems*, in: Proc. CONCUR'90, Amsterdam (J.C.M. Baeten & J.W. Klop, eds.), Springer LNCS 458, 1990, pp. 401-415.

[NSY91] X. NICOLLIN, J. SIFAKIS & S. YOVINE, *From ATP to timed graphs and hybrid systems*, in: Proc. REX Workshop on Real Time: Theory in Practice, Mook 1991 (J.W. de Bakker, C. Huizing, W.P. de Roever & G. Rozenberg, eds.), Springer LNCS 600, 1992, pp. 549-572.

[REI85] W. REISIG, *Petri nets*, EATCS monograph on TCS, Springer Verlag 1985.

[WIN87] G. WINSKEL, *Event structures*, in: Petri nets: applications and relationships to other models of concurrency, Bad Honnef 1986 (W. Brauer, W. Reisig & G. Rozenberg, eds.), Springer LNCS 255, 1987, pp. 325-392.

[YI90] WANG YI, *Real-time behaviour of asynchronous agents*. in: Proc. CONCUR'90, Amsterdam (J.C.M. Baeten & J.W. Klop, eds.), Springer LNCS 458, 1990, pp. 502-520.