Probabilistic and Prioritized Models of Timed CSP

Gavin Lowe

May 27, 1994

Abstract

In this paper we present two languages that are refinements of timed CSP [4]: a probabilistic language, and a fully deterministic language with a notion of priority.

In the first part of the paper we describe the deterministic language and its semantic model. The syntax is based upon that of timed CSP except some of the operators are refined so as to remove all nondeterminism; this produces prioritized operators. The semantics for our language represents a process as the set of possible behaviours for the process, where a behaviour models the priorities for different actions. A number of algebraic laws for our language are given and the model is illustrated with two examples.

In the second part of the paper, we extend the language by adding a probabilistic choice operator. We produce a semantic model for our language which gives the probabilities of different behaviours occurring, as well as modelling the relative priorities for events within a behaviour. The model is illustrated with an example of a communications protocol transmitting messages over an unreliable medium.

1 Introduction

Communicating sequential processes [5] is a language for reasoning about concurrent processes. This model has been extended [14, 15, 13] to include a treatment of timing information. Previous models have allowed nondeterminism; this has proved to be a useful tool in that it allows one to underspecify the behaviour of processes, and so maintain a high level of abstraction. However, previous models have failed to model the probabilities involved in nondeterministic choices. In this paper we aim to overcome this deficiency, and in doing so also produce a fully deterministic model with a notion of priority.

We have chosen to build our probabilistic model upon a deterministic model because it is our belief that in order to argue about probabilistic behaviours it is necessary to be able to predict precisely how the non-probabilistic parts behave in a given circumstance. If a language includes other forms of nondeterminism, besides probabilistic choice, then it is not possible to predict the probability of a particular behaviour occurring. For example, consider the question:

> What is the probability that the process $a \longrightarrow STOP \,\Box\, b \longrightarrow STOP$ performs an a if the environment is willing to perform either an a or a b at time 0?

In the standard models of timed CSP the external choice operator is underspecified, and so it is not possible to answer this question. We need to refine this operator in order to produce a deterministic version; this will then allow us to predict the behaviour of the process described above.

Some researchers have got around this problem by specifying that if a process is able to perform two or more separate actions then the choice is made by the environment. We avoid this because:

- we consider the environment to be a more passive entity than the process; it seems strange that an environment is able to choose between two actions whereas a process is not;
- this idea clashes with our intuition of a system (built out of smaller components) being in an environment consisting of a user who is willing to observe any event.

In more recent work [12], we have tried to combine probabilities with nondeterminism (in the sense of underspecification). The paper tried to represent processes as sets of probability functions, one probability function for each way of resolving the nondeterminism. However, we showed that none of the standard models of CSP—such as traces, failures or ready-sets—can be extended to cover both probabilities and underspecification; it seems very hard to combine the two phenomena.

We believe that a fully deterministic model is a useful thing in its own right because:

- nondeterminism can be considered a bad thing, in that a nondeterministic process is unpredictable and we would like programs that we write to always behave in a predictable way;
- the approach we will take will be to give certain actions higher priorities than other actions; this will give us a more powerful language for specifying processes.

Most previous probabilistic process algebras have used a probabilistic external choice operator, written say as ${}_p\Box_q$, such that $P\ {}_p\Box_q\ Q$ offers the environment a choice between the actions of P and Q; if the environment is willing to perform the actions of either, then P is chosen with probability p and Q is chosen with probability q (where $p + q = 1$). We choose to separate the two phenomena of external choice and probabilistic nondeterminism for we believe them to be orthogonal issues. Our language will include two deterministic external choice operators and a probabilistic internal choice operator. Having more operators produces a language that, while being harder to reason about, is easier to reason with.

Our deterministic external choice operators, which we will write as $[\!\!\Box$ and $\Box\!\!]$, will be the same as the operators ${}_1\Box_0$ and ${}_0\Box_1$: if the environment is willing to perform the actions of either P or Q, then $P\ [\!\!\Box\ Q$ will act like P, and $P\ \Box\!\!]\ Q$ will act like Q. Hence these are prioritized operators, and, in a sense, the prioritized external choice is the "limit" of a probabilistic external choice. The probabilistic external choice operator can be regained from the prioritized operators via the identity $P\ {}_p\Box_q\ Q = (P\ [\!\!\Box\ Q)\ {}_p\sqcap_q\ (P\ \Box\!\!]\ Q)$, where ${}_p\sqcap_q$ is a probabilistic internal choice operator.

We assume that the reader is familiar with the syntax and semantics of timed CSP, as described in, for example, [4].

The paper is split into three parts: the first two parts describe the deterministic and probabilistic models respectively; the third part sums up.

Part 1: In section 2 we introduce the syntax of our deterministic language; we illustrate the language with a couple of examples in section 3; in section 4 we describe our semantic space; we give semantic definitions for our operators in section 5.

Part 2: In section 6 we describe the additions to our syntax needed in the probabilistic model; in section 7 we describe our semantic space; in section 8 we give definitions for our operators; and in section 9 we illustrate our model with an example.

Part 3: In the final part we sum up and briefly discuss some more recent work on this subject.

Part I
The Deterministic Model

2 Syntax

We want to produce a language that is deterministic in the following sense: in each environment, there is only one way in which a given process can behave. In order to produce a deterministic language, we must first understand the ways in which nondeterminism can arise. Nondeterminism can arise in timed CSP in a number of ways:

Explicit nondeterminism: The process $P \sqcap Q$ chooses nondeterministically between the processes P and Q. In this paper we wish to deal with a completely deterministic model, and so we shall ban the nondeterministic choice operator from our syntax.

External choice: Consider the process $a \longrightarrow P \Box b \longrightarrow Q$: if the environment is willing to do either an a or a b at some time, then the choice is made nondeterministically.

Interleaving: Consider the process $a \longrightarrow P \mathbin{|||} b \longrightarrow Q$: if the environment is willing to allow exactly one of a and b, then the choice is made nondeterministically.

Hiding and renaming: Deterministic processes can sometimes be made nondeterministic by hiding or renaming. For example, if the process $a \longrightarrow P \Box b \longrightarrow Q$ is put in an environment that offers just a b at time 0, then the b will be performed. If however the process $(a \longrightarrow P \Box b \longrightarrow Q) \setminus a$ is put in the same environment then it will nondeterministically choose between performing the b or performing the a silently.

The last three forms can all be thought of as types of underspecification; in normal timed CSP we do not specify how the operators behave in the situations described. We shall refine our operators so as to overcome this underspecification.

2.1 Biased external choice

We define two external choice operators, $\overleftarrow{\Box}$ and $\overrightarrow{\Box}$. The left-biased choice $P \overleftarrow{\Box} Q$ will choose P if the environment is willing to do the first events of both P and Q (at some time). The right-biased choice $P \overrightarrow{\Box} Q$ will choose Q if the environment is willing to do the first events of both P and Q. For example, we can model a customer who is willing to accept a toffee, but would prefer a chocolate:

$$CUST \mathrel{\hat{=}} \mathit{chocolate} \overleftarrow{\Box} \mathit{toffee}$$

where we have written *chocolate* as an abbreviation for $\mathit{chocolate} \longrightarrow STOP$.

2.2 Parallel composition

Consider the process $(a \overleftarrow{\Box} b) \parallel (a \overrightarrow{\Box} b)$. If the environment offers both a and b at time 0, then the behaviour of the process is not fully specified. The left hand side wants to perform a, while the right hand side wants to perform b. The only sensible interpretation is that the process chooses nondeterministically between the a and the b. We are aiming to eliminate all nondeterminism so we must refine our syntax in some way: we define a left biased parallel operator $\overleftarrow{\parallel}$ which arbitrates in favour of its left hand argument; so, for example, $(a \overleftarrow{\Box} b) \overleftarrow{\parallel} (a \overrightarrow{\Box} b)$ will perform an a if the environment offers both a and b. We can consider the left hand side to be a master, and the right hand side to be a slave that will do whatever its master wants, if it can.

We can similarly define a right biased parallel operator $\mathbin{\parallel\!\triangleright}$ which arbitrates in favour of its right hand argument. For example, $(a \boxleft b) \mathbin{\parallel\!\triangleright} (a \boxright b)$ will perform a b if the environment offers both an a and a b.

For example, consider a vending machine that will dispense either chocolates or toffees as its environment requires, but would rather dispense toffees:

$$VMB \mathrel{\widehat{=}} \mathit{chocolate} \boxright \mathit{toffee}$$

If we put this in parallel with the customer who prefers chocolates, with the customer acting as the master, then the customer gets what he wants:

$$CUST \mathbin{\triangleleft\!\parallel} VMB = \mathit{chocolate} \boxleft \mathit{toffee}$$

If however we make the machine the master, then it gets its way:

$$CUST \mathbin{\parallel\!\triangleright} VMB = \mathit{chocolate} \boxright \mathit{toffee}$$

2.3 Interleaving

We define a left biased interleave operator, $\mathbin{\triangleleft\!|\!|\!|}$, such that if the environment is willing to do events of P or of Q (but not both) then $P \mathbin{\triangleleft\!|\!|\!|} Q$ performs the events of P. For example:

- if a single a is offered then $a \longrightarrow P \mathbin{\triangleleft\!|\!|\!|} a \longrightarrow Q$ will perform the a on the left;
- $(a \mathbin{\triangleleft\!|\!|\!|} b) \mathbin{\triangleleft\!\parallel} (a \boxright b)$ will perform an a if an a and a b are offered at the same time;
- $(a \mathbin{\triangleleft\!|\!|\!|} b) \mathbin{\parallel\!\triangleright} (a \boxright b)$ will perform a b if an a and a b are offered at the same time, since the right hand side is the master and it prefers the b.
- A greedy customer would like both a chocolate and a toffee, but if he can have only one he would prefer a chocolate:

  $$GCUST \mathrel{\widehat{=}} \mathit{chocolate} \mathbin{\triangleleft\!|\!|\!|} \mathit{toffee}$$

  When he is placed in parallel with the biased vending machine, with him as the master, he gets just a chocolate since the vending machine is only willing to dispense one sweet:

  $$GCUST \mathbin{\triangleleft\!\parallel} VMB = \mathit{chocolate} \boxleft \mathit{toffee}$$

Similarly, we define a right-biased interleaving operator $\mathbin{|\!|\!|\!\triangleright}$ that arbitrates in favour of its right-hand argument.

2.4 Complete syntax

The complete syntax for deterministic timed CSP (DTCSP) is as follows

$$\begin{aligned} P ::= {} & STOP \mid SKIP \mid WAIT\ t \mid a \xrightarrow{t} P \mid P \boxleft P \mid P \boxright P \mid P \mathbin{\triangleleft\!\parallel} P \mid P \mathbin{\parallel\!\triangleright} P \mid \\ & P\ {}^{A}\!\!\mathbin{\triangleleft\!\parallel}^{B}\ P \mid P\ {}^{A}\!\!\mathbin{\parallel\!\triangleright}^{B}\ P \mid P \mathbin{\triangleleft\!|\!|\!|} P \mid P \mathbin{|\!|\!|\!\triangleright} P \mid P \fatsemi P \mid WAIT\ t\,; P \mid P \setminus A \mid \\ & f(P) \mid \mu X \bullet F(X) \end{aligned}$$

where t ranges over the set $TIME$ of times, which we take to be positive real numbers; a ranges over some alphabet Σ of events; A and B range over $\mathbb{P}\,\Sigma$; f over $\Sigma \to \Sigma$; and F over functions from process names to processes.

2.5 The effect of hiding

Consider the process $P \mathrel{\widehat{=}} (a \mathbin{\square\!\!\!\lfloor}\, b) \setminus a$. It is interesting to ask whether this process can ever perform a b. The process P certainly prefers to perform an a (silently) to a b. In a previous paper [6] we took the view that the environment would always be willing to perform no events; hence P could never perform a b since it would always choose to perform a silent a. This assumption produces a model which, while self-consistent, is extremely complicated and contains a number of unusual and undesirable features.

In this paper we adopt the view that the environment is *not* always willing to idle. Then the process P *is* able to perform a b, but only if its environment is not willing to perform the empty bag of events. Consider for example the process $b \mathbin{\triangleleft\!\!\|} P$. The left hand side of this prefers to perform a b than to idle; it is the master and so it forces P to perform the b even though P would prefer to perform a silent a.

3 Examples

3.1 A shared resource

In this section we give an example which illustrates one use of the prioritized operators: we use the operators to perform arbitration when a resource is required by two different users. We shall write $\square$, $\|$ and $|||$ to denote external choice, parallel or interleave operators that could be either left- or right-biased.

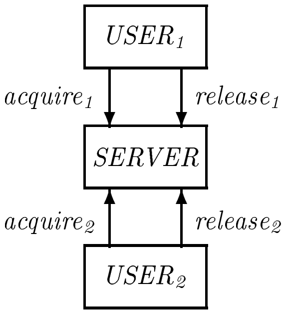

Figure 1: A shared resource

Two users, $USER_1$ and $USER_2$, share a single resource, $SERVER$, as in figure 1. The users may acquire the server and then release it:

$$USER_1 \mathrel{\widehat{=}} \mu X \bullet acquire_1 \longrightarrow release_1 \longrightarrow X \qquad USER_2 \mathrel{\widehat{=}} \mu X \bullet acquire_2 \longrightarrow release_2 \longrightarrow X$$

The server can be acquired or released by either user, but cannot be acquired by one user until it has been released by the other.

$$\begin{aligned} SERVER \mathrel{\widehat{=}} \mu X \bullet\ & acquire_1 \longrightarrow release_1 \longrightarrow X \\ & \square \\ & acquire_2 \longrightarrow release_2 \longrightarrow X \end{aligned}$$

The system consists of the two users in parallel without communication (i.e. interleaved), in parallel with the server.

$$SYSTEM \mathrel{\widehat{=}} (USER_1 \mathbin{|||} USER_2) \mathbin{\|} SERVER$$

We also want to insist that if both users try to acquire the server at the same time then $USER_1$ acquires it. We examine two ways of doing this.

1. We specify that the server would rather be acquired by $USER_1$ than $USER_2$:

$$SERVER \mathrel{\widehat{=}} \mu X \bullet acquire_1 \longrightarrow release_1 \longrightarrow X \mathbin{\Box\!\!\!\mid} acquire_2 \longrightarrow release_2 \longrightarrow X$$

   We must then make the server the master to ensure that it is able to make the choice:

$$SYSTEM \mathrel{\widehat{=}} SERVER \mathbin{\triangleleft\!\!\|} (USER_1 \mathbin{|||} USER_2)$$

2. We give $USER_1$ preference over $USER_2$ by making it the master of the interleaving, and giving the users mastery over the server:

$$SYSTEM \mathrel{\widehat{=}} (USER_1 \mathbin{\triangleleft\!\!|||} USER_2) \mathbin{\triangleleft\!\!\|} SERVER$$

3.2 An interrupt mechanism

We consider now the example of an interrupt mechanism, described in [2] and [1] and illustrated in figure 2. A counter can normally continually perform the events *up* and *down*. If, however, the event

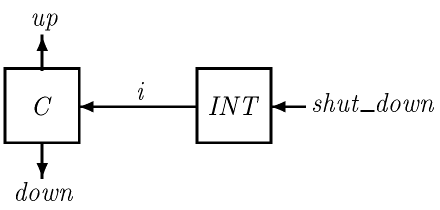

Figure 2: A counter with interrupt mechanism

shut_down occurs, then it should be interrupted via the internal event i. In an unprioritized model the definition would be

$$\begin{aligned}
SYS &\mathrel{\widehat{=}} (C_0 \; {}^{X}\|^{Y} \; INT) \setminus i \\
INT &\mathrel{\widehat{=}} shut_down \longrightarrow i \longrightarrow STOP \\
C_0 &\mathrel{\widehat{=}} up \longrightarrow C_1 \Box i \longrightarrow STOP \\
C_{n+1} &\mathrel{\widehat{=}} (up \longrightarrow C_{n+2} \Box down \longrightarrow C_n) \Box i \longrightarrow STOP
\end{aligned}$$

where the alphabets are given by $X \mathrel{\widehat{=}} \{up, down, i\}$, $Y \mathrel{\widehat{=}} \{i, shut\_down\}$.
It should be obvious that this could perform the trace $\langle up, down, shut\_down, up, down\rangle$ — C can choose to ignore the event i offered by INT, in favour of *up*s and *down*s. We can get around this by giving the i a higher priority than the *up* and the *down*:

$$\begin{aligned}
C_0 &\mathrel{\widehat{=}} up \longrightarrow C_1 \mathbin{\Box\!\!\!\mid} i \longrightarrow STOP \\
C_{n+1} &\mathrel{\widehat{=}} (up \longrightarrow C_{n+2} \Box down \longrightarrow C_n) \mathbin{\Box\!\!\!\mid} i \longrightarrow STOP
\end{aligned}$$

where the $\Box$ could be either a left- or right-biased external choice. Now the i will be performed as soon as it is offered, and C will be interrupted as required.

4 Semantics

In this section we develop a semantic model for our language. We begin by describing how we want to model a typical behaviour of a process; we then present some notation, before producing the semantic model itself, which will represent a process by the set of all behaviours that it can perform.

4.1 Behaviours

As in most models of concurrency, we want our representation of a behaviour (or observation) of a process to record the events performed. Since we are interested in the different priorities given to different actions, we also want to include some representation of these priorities. It will ease our notation to also include the time at which the observation ends. Our model of a behaviour will therefore consist of three parts: the time up until which the process is observed, the events that it performs and the priorities given to different actions.

The trace of a process is the collection of timed events that it performs. In standard timed CSP the traces $\langle(0,a),(0,b)\rangle$ and $\langle(0,b),(0,a)\rangle$ are treated as distinct. In this paper we want to associate these; otherwise when we come to consider probabilities we will experience problems. Consider, for example, the process $a \mathbin{\#\!\!\!\!\leftarrow} b$: if the environment is willing to perform an a and a b at time 0 then this can perform the trace $\langle(0,a),(0,b)\rangle$ with probability one and can also perform the trace $\langle(0,b),(0,a)\rangle$ with probability one: our probabilities will not sum to one. In our model we shall say that this process — in this environment — performs the bag $\{\!|a,b|\!\}$ at time zero with probability one.

We shall represent traces as functions from a time interval to bags of events. The space TT of timed traces is defined by

$$TT \mathrel{\widehat=} \{s : TIME \nrightarrow \mathrm{bag}\,\Sigma \mid \mathrm{dom}\, s \in TINT\}$$

where $TINT$ is the space of finite time intervals (open, closed, or half open). We think of $s(t)$ as being the bag of events performed at time t. Both of the traces described in the previous paragraph are represented by $\lambda\, t \bullet \text{if } t = 0 \text{ then } \{\!|a,b|\!\} \text{ else } \{\!||\!\}$. For ease of notation, we shall often write traces as sequences within the brackets $\langle$ and $\rangle$, so the above trace will be denoted by either $\langle(0,a),(0,b)\rangle$ or $\langle(0,b),(0,a)\rangle$.

We say that a process *offers* a particular bag of events if it is willing to perform that bag, i.e. if it offers the bag to the environment. We define the space of offers OFF by

$$OFF \mathrel{\widehat=} TIME \times \mathrm{bag}\,\Sigma$$

The pair (t,χ) represents the bag of events χ being offered at time t. We shall write v, w, etc. for typical members of OFF, and χ, ψ, etc. for typical members of $\mathrm{bag}\,\Sigma$.

Note: It is normal to consider a function from type a to type b to be of type $\mathbb{P}(a \times b)$. Using this identification, we can consider a timed trace to be of type $\mathbb{P}(TIME \times \mathrm{bag}\,\Sigma)$, i.e. a trace is simply a collection of offers. We will make use of this to simplify our notation.

A process will often be willing to offer more than one particular bag of events. It will then have some preference as to which bag of events it would rather perform. For example, the process $a \mathbin{\#\!\!\!\!\leftarrow} b$ initially offers the bags $\{\!|a,b|\!\}$, $\{\!|a|\!\}$, $\{\!|b|\!\}$, and $\{\!||\!\}$, and prefers $\{\!|a,b|\!\}$ to $\{\!|a|\!\}$, prefers $\{\!|a|\!\}$ to $\{\!|b|\!\}$, and prefers $\{\!|b|\!\}$ to $\{\!||\!\}$. We want to model the order of preference of offers. We define the space $OFFREL$ of offer relations to be those relations $\sqsubseteq$ of type $OFF \times OFF$ satisfying the following conditions:

1. $(t,\chi) \sqsubseteq (t',\psi) \Rightarrow t = t'$ (only actions at the same time are comparable);
2. $w \sqsubseteq w' \wedge w' \sqsubseteq w'' \Rightarrow w \sqsubseteq w''$ (transitivity);
3. $w \sqsubseteq w' \wedge w' \sqsubseteq w \Rightarrow w = w'$ (antisymmetry);
4. $w \in \mathrm{items} \sqsubseteq \Rightarrow w \sqsubseteq w$ (reflexivity on $\mathrm{items} \sqsubseteq$);
5. $(t,\chi),(t,\psi) \in \mathrm{items} \sqsubseteq \Rightarrow (t,\chi) \sqsubseteq (t,\psi) \vee (t,\psi) \sqsubseteq (t,\chi)$ (totality on $\mathrm{items} \sqsubseteq$);
6. $\{t \mid \exists\, \chi \bullet (t,\chi) \in \mathrm{items} \sqsubseteq\} \in TINT$ (time domain is a time interval);

where $\mathrm{items} \sqsubseteq$ is the set of all bags of events offered by the process:

$$\mathrm{items} \sqsubseteq \mathrel{\widehat=} \{w \mid \exists\, v \bullet w \sqsubseteq v \vee v \sqsubseteq w\}$$

Informally, if $v \sqsubseteq w$ then the process would rather perform w than v. For example, $a \mathrel{\leftarrow\!\!|\!|\!|} b$ has offer relation with $(\mathit{0},\ \{\!|\,|\!\}) \sqsubseteq (\mathit{0},\ \{\!|b|\!\}) \sqsubseteq (\mathit{0},\ \{\!|a|\!\}) \sqsubseteq (\mathit{0},\ \{\!|a, b|\!\})$. We introduce the following shorthands:

$$v \sqsubset w \Leftrightarrow v \sqsubseteq w \wedge v \neq w \qquad v \sqsupseteq w \Leftrightarrow w \sqsubseteq v \qquad v \sqsupset w \Leftrightarrow w \sqsubseteq v \wedge v \neq w$$

An environmental offer is the collection of timed events that the process is offered by the environment; more formally, it is a set of offers, i.e. a set of type $\mathbb{P}(\mathit{OFF})$. We let EOFF be the set of all environmental offers and write Ω for a typical member. We shall discuss environmental offers more fully after we have introduced some notation.

A behaviour will be a triple of type $\mathit{TIME} \times \mathit{OFFREL} \times \mathit{TT}$. The behaviour $(\tau, \sqsubseteq, s)$ will represent an observation up until time τ, where trace s is observed, and where the offer relation is $\sqsubseteq$. We shall discuss which behaviours are possible after we have introduced some notation.

4.2 Notation

Our notation is an extension of the notation of [4].

Bag notation

Our notation for bags is as follows: we write $b.e$ for the number of times element e occurs in bag b; $e \in b$ is true iff $b.e > \mathit{0}$. We have a number of operations on bags:

$$\begin{array}{ll} (b_1 \cup b_2).e = b_1.e \sqcup b_2.e & \quad (b_1 \cap b_2).e = b_1.e \sqcap b_2.e \\ (b_1 - b_2).e = (b_1.e - b_2.e) \sqcup \mathit{0} & \quad (b_1 \uplus b_2).e = b_1.e + b_2.e \end{array}$$

where the operators $\sqcup$ and $\sqcap$ return the maximum and minimum of their arguments respectively. Bag enumerations and bag comprehensions are written within bag brackets $\{\!|$ and $|\!\}$. If a particular value of a bound variable occurs more than once in a bag comprehension, then the corresponding term occurs more than once.

We shall write $\sum \left\{\!\left| f(x) \mid p(x) \right|\!\right\}$ to represent the sum of the $f(x)$, where the sum is taken over all x such that $p(x)$ holds.

Notation for offers

We define projection functions that select the time and event bag components of an offer:

$$I(t, \chi) \mathrel{\widehat=} t \qquad \Sigma(t, \chi) \mathrel{\widehat=} \chi$$

We define restriction, hiding and renaming operators on offers:

$$\begin{array}{rcl} (t, \chi) \upharpoonright X & \widehat= & (t,\ \{\!|a \in \chi \mid a \in X|\!\}) \\ (t, \chi) \setminus X & \widehat= & (t,\ \{\!|a \in \chi \mid a \notin X|\!\}) \\ g(t, \chi) & \widehat= & (t, g\chi) \end{array}$$

We define partial union, bag union and bag difference operators on offers:

$$(t, \chi) \cup (t, \psi) \mathrel{\widehat=} (t, \chi \cup \psi) \qquad (t, \chi) \uplus (t, \psi) \mathrel{\widehat=} (t, \chi \uplus \psi) \qquad (t, \chi) - (t, \psi) \mathrel{\widehat=} (t, \chi - \psi)$$

Notation for traces

The interval function *times* returns the set of times at which events occur:

$$\mathit{times}\ s \mathrel{\widehat=} \{t \mid s(t) \neq \{\!|\,|\!\}\}$$

This contrasts with the function I which returns the set of all times in the domain of a trace:

$$Is \mathrel{\widehat=} \operatorname{dom} s$$

We define *begin* and *end* operators which return the times of the first and last events of a trace; *times s* will always be finite, so *begin* and *end* will return the minimum and maximum of *times s*, respectively:

$$\mathit{begin}\ s \mathrel{\widehat=} \begin{cases} \infty & \text{if } \mathit{times}\ s = \{\} \\ \min(\mathit{times}\ s) & \text{otherwise} \end{cases} \qquad \mathit{end}\ s \mathrel{\widehat=} \begin{cases} -\infty & \text{if } \mathit{times}\ s = \{\} \\ \max(\mathit{times}\ s) & \text{otherwise} \end{cases}$$

The *first* and *last* operators return the bags of initial or final events of a non-empty trace:

$$\mathit{first}\ s \mathrel{\widehat=} s(\mathit{begin}\ s) \qquad \mathit{last}\ s \mathrel{\widehat=} s(\mathit{end}\ s)$$

The alphabet function Σ returns the set of (untimed) events from a trace:

$$\Sigma s \mathrel{\widehat=} \{a \mid \exists\, t \bullet a \in s(t)\}$$

The during operator, $\uparrow$, returns the portion of a trace that occurs during some time interval:

$$s \uparrow I \mathrel{\widehat=} \{t \mapsto s(t) \mid t \in I\}$$

We use this to define before, strictly before, after, strictly after and at operators, $\trianglelefteq$, $\triangleleft$, $\trianglerighteq$, $\triangleright$ and $\uparrow$:

$$\begin{array}{ll} s \trianglelefteq t \mathrel{\widehat=} s \uparrow [0, t] & s \triangleleft t \mathrel{\widehat=} s \uparrow [0, t) \\ s \trianglerighteq t \mathrel{\widehat=} s \uparrow [t, \infty) & s \triangleright t \mathrel{\widehat=} s \uparrow (t, \infty) \\ s \uparrow t \mathrel{\widehat=} s \uparrow [t, t] & \end{array}$$

We define a partial concatenation operator on traces:

$$s_1 \frown s_2 = s_1 \cup s_2 \qquad \text{if } Is_1 \cap Is_2 = \{\} \wedge Is_1 \cup Is_2 \in \mathit{TINT}$$

We define two temporal shift operators:

$$s + t \mathrel{\widehat=} \{t + t' \mapsto s(t') \mid t' \in Is\} \qquad s - t \mathrel{\widehat=} \{t' - t \mapsto s(t') \mid t' \in Is \wedge t' \geqslant t\}$$

We define restriction and hiding operators:

$$s \upharpoonright X \mathrel{\widehat=} \{(t, \chi \upharpoonright X) \mid (t, \chi) \in s\} \qquad s \setminus X \mathrel{\widehat=} \{(t, \chi \setminus X) \mid (t, \chi) \in s\}$$

Notation for offer relations

Recall the definition of the function items, which returns the set of all offers of an offer relation:

$$\text{items} \sqsubseteq \mathrel{\widehat=} \{w \mid \exists\, v \bullet w \sqsubseteq v \vee v \sqsubseteq w\}$$

Most of the rest of our notation for offer relations is similar to our notation for traces.

$$\begin{array}{rl} I\sqsubseteq \mathrel{\widehat=} \{t \mid \exists\, \chi \bullet (t, \chi) \in \text{items} \sqsubseteq\} & \text{duration} \\ \mathit{begin} \sqsubseteq \mathrel{\widehat=} \begin{cases} \infty & \text{if } I\sqsubseteq = \{\} \\ \inf(I\sqsubseteq) & \text{otherwise} \end{cases} & \text{begin} \\ \mathit{end} \sqsubseteq \mathrel{\widehat=} \begin{cases} -\infty & \text{if } I\sqsubseteq = \{\} \\ \sup(I\sqsubseteq) & \text{otherwise} \end{cases} & \text{end} \\ \sqsubseteq \uparrow I \mathrel{\widehat=} \sqsubseteq' \text{ where } (t, \chi) \sqsubseteq' (t, \psi) \Leftrightarrow t \in I \wedge (t, \chi) \sqsubseteq (t, \psi) & \text{during} \\ \sqsubseteq \trianglelefteq t \mathrel{\widehat=} \sqsubseteq \uparrow [0, t] & \text{before} \\ \sqsubseteq \triangleleft t \mathrel{\widehat=} \sqsubseteq \uparrow [0, t) & \text{strictly before} \\ \sqsubseteq \trianglerighteq t \mathrel{\widehat=} \sqsubseteq \uparrow [t, \infty) & \text{after} \\ \sqsubseteq \triangleright t \mathrel{\widehat=} \sqsubseteq \uparrow (t, \infty) & \text{strictly after} \\ \sqsubseteq + t \mathrel{\widehat=} \sqsubseteq' \text{ where } (t + t', \chi) \sqsubseteq' (t + t', \psi) \Leftrightarrow (t', \chi) \sqsubseteq (t', \psi) & \text{temporal shift forwards} \\ \sqsubseteq - t \mathrel{\widehat=} \sqsubseteq' \text{ where } (t' - t, \chi) \sqsubseteq' (t' - t, \psi) \Leftrightarrow (t', \chi) \sqsubseteq (t', \psi) \wedge t' \geqslant t & \text{temporal shift back} \\ \sqsubseteq_1 \frown \sqsubseteq_2 \mathrel{\widehat=} \sqsubseteq_1 \cup \sqsubseteq_2 \text{ if } I\sqsubseteq_1 \cap I\sqsubseteq_2 = \{\} \wedge I\sqsubseteq_1 \cup I\sqsubseteq_2 \in \mathit{TINT} & \text{concatenation} \end{array}$$

We define an operator $\otimes$, which we will use for representing offer relation: $I \otimes \langle \chi_0, \chi_1, \ldots, \chi_{n-1} \rangle$ represents the offer relation $\sqsubseteq$, such that for all times t during I, $(t, \chi_0) \sqsupset (t, \chi_1) \sqsupset (t, \chi_2) \sqsupset \ldots \sqsupset (t, \chi_{n-1})$.

$$\forall I : TINT \,;\, d : \mathrm{seq}(\mathrm{bag}\,\Sigma) \bullet I \otimes d = \sqsubseteq$$
$$\text{where } (t, \chi) \sqsubseteq (t', \psi) \Leftrightarrow t = t' \in I \wedge \exists\, i, j \bullet 0 \leqslant i \leqslant j < \#d \wedge d(i) = \psi \wedge d(j) = \chi$$

Notation for environmental offers

We define temporal restriction operators on environmental offers in the obvious way. If Ω is an environmental offer, we define

$$\begin{array}{lll} \Omega \uparrow I \mathrel{\widehat=} \{(t, \chi) \in \Omega \mid t \in I\} & \Omega \lhd t \mathrel{\widehat=} \Omega \uparrow [0, t) & \Omega \unlhd t \mathrel{\widehat=} \Omega \uparrow [0, t] \\ \Omega \rhd t \mathrel{\widehat=} \Omega \uparrow (t, \infty) & \Omega \unrhd t \mathrel{\widehat=} \Omega \uparrow [t, \infty) & \Omega \uparrow t \mathrel{\widehat=} \Omega \uparrow [t, t] \end{array}$$

We denote the maximum elements of Ω under $\sqsubseteq$ by $\sqcup_{\sqsubseteq} \Omega$:

$$\sqcup_{\sqsubseteq} \Omega \mathrel{\widehat=} \{(t, \chi) \in \Omega \cap \mathrm{items} \sqsubseteq \mid \forall\, \psi \bullet (t, \psi) \in \mathrm{items} \sqsubseteq \cap\, \Omega \Rightarrow (t, \psi) \sqsubseteq (t, \chi)\}$$

Note that $\sqcup_{\sqsubseteq} \Omega$ is a set of offers, one offer for each time during the duration of Ω, and so can be thought of as a trace — namely the trace where at each instant the element of Ω that is maximal under $\sqsubseteq$ is performed. This will be the trace that a process with offer relation $\sqsubseteq$ will perform when placed in an environment Ω.

4.3 Possible behaviours

Only certain behaviours $(\tau, \sqsubseteq, s)$ are possible. We want to limit our attention to those that satisfy a number of healthiness conditions which express some of our intuitions about how a process should behave. We introduce the space *BEH* of possible behaviours:

Definition 4.1 (Possible behaviours): The space *BEH* is defined to be those triples $(\tau, \sqsubseteq, s)$ of type $TIME \times OFFREL \times TT$ satisfying the following conditions:

1. $I \sqsubseteq = Is = [0, \tau]$
2. $\forall\, t \leqslant \tau \bullet s \uparrow t \in \mathrm{items} \sqsubseteq$
3. $t_0 < t_1 \wedge \{(t, \chi) \mid t \in (t_0, t_1)\} \subseteq \mathrm{items} \sqsubseteq \Rightarrow (t_0, \chi) \uplus s \uparrow t_0 \sqsupseteq s \uparrow t_0$
4. $(t, \chi) \in \mathrm{items} \sqsubseteq \wedge\, \psi \subseteq \chi \Rightarrow (t, \psi) \in \mathrm{items} \sqsubseteq$
5. $(t < t' \wedge \forall\, t'' \in [t, t') \bullet (t'', \chi) \in \mathrm{items} \sqsubseteq) \Rightarrow (t', \chi) \in \mathrm{items} \sqsubseteq$

$\diamondsuit$

We discuss the five conditions in turn:

1. If a process is observed up until time τ, then no events will be seen to be performed or offered after time τ.
2. A process can only perform events that it is willing to offer to parallel processes.
3. If a process is willing to perform a bag of events χ throughout some open time interval beginning at t_0, then at t_0 it could have performed this bag as well as what it did perform; further it would have preferred to perform χ and what it did perform to just what it did perform. This condition is necessary to avoid processes such as the one that offers an a during $(0, 1)$; if the environment offers an a from time 0 onwards, then there is no sensible choice as to when the a should be performed.

4. The offers of a process are subbag closed.

5. If a process offers a bag χ at all times just before t', then it also offers χ at t'.

In section 4.5 we will also show that the behaviours satisfy the following finite speed conditions:

- The process can only perform a finite number of events in a finite time:

$$\#s < \infty$$

- The offer relation changes shape a finite number of times:

$$\begin{array}{l} \exists\, k : \mathbb{N} \,;\, I_0, \ldots, I_{k-1} \in TINT \bullet \\ \quad I_0, \ldots I_{k-1} \text{ partition } [0, \tau] \\ \quad \wedge \forall\, i : 0 \,..\, k-1 \,;\, t, t' \in I_i \,;\, \chi, \psi \in \text{bag}\,\Sigma \bullet (t, \chi) \sqsubseteq (t, \psi) \Leftrightarrow (t', \chi) \sqsubseteq (t', \psi) \end{array}$$

  There is a finite number of time intervals $I_0, \ldots, I_{k-1}$ such that the offer relation does not change shape during each interval I_i.

Using these conditions, we can show that the empty bag is always offered:

Theorem 4.2: $\forall (\tau, \sqsubseteq, s) \in BEH \bullet \forall\, t \in [0, \tau] \bullet (t, \{\!|\,|\!\}) \in \text{items} \sqsubseteq$. ♡

4.4 Environmental offers

We think of the environment for a process P as being a process Q running in parallel with P; we think of the environmental offer as being the set of offers of Q. We shall say that a behaviour $(\tau, \sqsubseteq, s)$ is compatible with an environmental offer Ω (of type $\mathbb{P}(OFF)$) if $(\tau, \sqsubseteq, s)$ could have resulted from the environment offering Ω.

Definition 4.3: The behaviour $(\tau, \sqsubseteq, s)$ is compatible with the environmental offer Ω, written $(\tau, \sqsubseteq, s)$ *compat* Ω, if:

1. $I\Omega = [0, \tau]$
2. $\forall\, t \bullet s \uparrow t = \sqcup_{\sqsubseteq} \Omega \uparrow t$
3. $(\forall\, t \in (t_0, t_1) \bullet (t, \chi) \in \Omega) \Rightarrow (t_0, \chi) \uplus s \uparrow t_0 \in \Omega$

◇

These conditions state that:

1. The duration of the environmental offer is the same as the duration of the behaviour;
2. At all times, the process performs the element of the environmental offer that is maximal under its offer relation — in other words, the process picks the offer that it prefers;
3. If a bag of events χ is offered throughout some open time interval beginning at t_0, then at t_0 the environment must have offered χ along with what was performed at t_0. This condition is necessary to avoid an environment such as the one that offers an a during the period $(0, 1]$; if a process that is willing to perform an a from time 0 onwards is placed in this environment, then there is no sensible choice as to when the event should be performed.

Of these, condition 2 is perhaps the most important. It describes the way that a process chooses the events it performs. At each instant the environment is willing to perform any one of a number of bags of events; the process takes its pick from these by choosing the bag that is strongest under its offer relation.

We can now define the space $EOFF$ of environmental offers:

$$EOFF \mathrel{\widehat{=}} \{\Omega : \mathbb{P}(OFF) \mid \exists (\tau, \sqsubseteq, s) \in BEH \bullet (\tau, \sqsubseteq, s) \; compat \; \Omega\}$$

A collection of offers is an environmental offer if it is compatible with *some* behaviour.

In general, we will allow the environmental offer to be a function of the observed behaviour. This fits in with our intuition of the environment for process P being another process, Q, running in parallel with P; different behaviours of P will cause Q to act in different ways and so will cause different environmental offers in the future. In general, it is enough to allow the environment to depend upon the offer relation of the process. When we want to stress that environment Ω is a function of the offer relation $\sqsubseteq$, we will write $\Omega(\sqsubseteq)$.

4.5 The semantic space DTM_B

We are now ready to define our semantic space. Firstly, we give a name to the space of sets of prioritized behaviours

$$TS_B \mathrel{\widehat{=}} \mathbb{P}(BEH)$$

TS_B is the space of timed sets using biases. We define the space DTM_B (the Deterministic Timed Model using Biases) to be those sets A of type TS_B satisfying a number of healthiness conditions. Intuitively, the set A represents a process that can perform any of the behaviours in A. The set A must obey the following axioms:

1. $\forall \tau \geqslant 0 \bullet \exists\, n(\tau) \bullet (\tau, \sqsubseteq, s) \in A \Rightarrow \# s \leqslant n(\tau)$
2. $\forall \tau \geqslant 0 \bullet \exists\, n(\tau) \bullet (\tau, \sqsubseteq, s) \in A \Rightarrow \exists\, k \leqslant n(\tau) \, ; I_0, \ldots, I_{k-1} \in TINT \bullet$
   $I_0, \ldots I_{k-1}$ partition $[0, \tau]$
   $\wedge \forall i : 0 \,..\, k-1 \, ; t, t' \in I_i \, ; \chi, \psi \in \operatorname{bag} \Sigma \bullet (t, \chi) \sqsubseteq (t, \psi) \Leftrightarrow (t', \chi) \sqsubseteq (t', \psi)$
3. $(\tau, \sqsubseteq, s) \in A \wedge (t, \chi) \in \operatorname{items} \sqsubseteq \Rightarrow (t, \sqsubseteq \trianglelefteq t, s \lhd t ^\frown (t, \chi)) \in A$
4. $\forall \Omega : OFFREL \rightarrow EOFF \bullet \exists ! \sqsubseteq \bullet (end\, \Omega, \sqsubseteq, \sqcup_{\sqsubseteq} \Omega(\sqsubseteq)) \in A$

We discuss each of these axioms in turn:

1. The number of events that a process can perform in a finite time is bounded.
2. The number of times at which an offer relation can change in a finite time is bounded.
3. A process is able to perform any bag of events that it offers.
4. The process is deterministic: given the way the environmental behaves, there is a unique offer relation that it can have; it will perform those members of the environmental offer that are maximal with respect to this offer relation.

The following law can be deduced from the axioms: if a process can have a particular behaviour, then it can perform any prefix of that behaviour:

Theorem 4.4: $(\tau, \sqsubseteq, s) \in A \wedge \tau' \leqslant \tau \Rightarrow (\tau', \sqsubseteq \trianglelefteq \tau', s \trianglelefteq \tau') \in A \wedge (\tau', \sqsubseteq \trianglelefteq \tau', s \lhd \tau' ^\frown (\tau', \{\!|\,|\!\})) \in A$. ♡

4.6 Semantic functions

We shall define a function $\mathcal{A}_{DT} : DTCSP \to DTM_B$ such that $\mathcal{A}_{DT}[\![P]\!]$ gives the set of possible behaviours of process P. In the following section we give definitions for $\mathcal{A}_{DT}$ for each of the constructs of our language; in most cases the crux of the definition will be the description of how the offer relation of a composite process results from the offer relations of its subcomponents. The definitions were proved sound (in that they respect the axioms) in [7]. We will state a number of laws that can be shown to hold of our processes, and also show why a few laws that hold in the unprioritized models do not hold in this model. Most of the laws were proved sound in [6] or [7].

5 Definitions of the operators

5.1 *STOP*

The process $STOP$ always performs the empty trace and offers nothing:

$$\mathcal{A}_{DT}[\![STOP]\!] \mathrel{\widehat=} \{(\tau, [0,\tau] \otimes \langle \{\!|\!\}\rangle, \prec\succ) \mid \tau \in TIME\}$$

5.2 *WAIT*

The process $WAIT\ t$ behaves as follows:

- if the environment does not offer $\checkmark$ at or after t, then it performs the empty trace, and offers $\checkmark$ from t onwards;
- if $\checkmark$ is offered by the environment at or after t then it is immediately performed; $\checkmark$ will be offered to parallel processes from time t until it is performed.

This gives the following definition:

$$\begin{aligned}\mathcal{A}_{DT}[\![WAIT\ t]\!] \mathrel{\widehat=} &\{(\tau, [0,\tau] \otimes \langle \{\!|\!\}\rangle, \prec\succ) \mid \tau < t\}\\ &\cup\{(\tau, [0,t) \otimes \langle \{\!|\!\}\rangle ^\frown [t,\tau] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\!\}\rangle, \prec\succ) \mid \tau \geqslant t\}\\ &\cup\{(\tau, [0,t) \otimes \langle \{\!|\!\}\rangle ^\frown [t,t'] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\!\}\rangle ^\frown (t',\tau] \otimes \langle \{\!|\!\}\rangle, \langle (t',\checkmark)\rangle) \mid t \leqslant t' \leqslant \tau\}\end{aligned}$$

5.3 *SKIP*

$SKIP$ is equivalent to $WAIT\ 0$, so we have the following definition:

$$\begin{aligned}\mathcal{A}_{DT}[\![SKIP]\!] \mathrel{\widehat=} &\{(\tau, [0,\tau] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\!\}\rangle, \prec\succ) \mid \tau \in TIME\}\\ &\cup\{(\tau, [0,t] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\!\}\rangle ^\frown (t,\tau] \otimes \langle \{\!|\!\}\rangle, \langle (t,\checkmark)\rangle) \mid t \leqslant \tau\}\end{aligned}$$

5.4 Prefixing

The process $a \xrightarrow{0} P$ should offer an a until it is performed, and then act like P. In order for this to fit with our intuition of causality, we insist that P is unable to perform any events at time 0 (i.e., the $a \xrightarrow{0}$ operator is a *partial* function, defined for only such P).

$$\begin{aligned}&\mathcal{A}_{DT}[\![a \xrightarrow{0} P]\!] \mathrel{\widehat=}\\ &\quad\{(\tau, [0,\tau] \otimes \langle \{\!|a|\!\}, \{\!|\!\}\rangle, \prec\succ)\}\\ &\quad\cup\{(\tau, [0,t] \otimes \langle \{\!|a|\!\}, \{\!|\!\}\rangle ^\frown \sqsubseteq_P + t, (t,a) ^\frown s_P + t) \mid (\tau - t, 0 \otimes \langle \{\!|\!\}\rangle ^\frown \sqsubseteq_P, s_P) \in A_P \wedge \tau \geqslant t\}\end{aligned}$$

where here and henceforth $A_P \mathrel{\widehat=} \mathcal{A}_{DT}[\![P]\!]$. We define the general prefix operator by $a \xrightarrow{t} P \mathrel{\widehat=} a \xrightarrow{0} WAIT\ t\,; P$. We will write $a \longrightarrow P$ as a shorthand for $a \xrightarrow{\delta} P$.

5.5 External choice

Consider the process $P \boxempty Q$. We want to derive a definition for the offer relation of $P \boxempty Q$ in terms of the offer relations of P and Q. We begin by considering an example. Suppose P has offer relation $\sqsubseteq_P$ and Q has offer relation $\sqsubseteq_Q$, with $\{\!|a|\!\} \sqsupset_P \{\!|\,|\!\} \sqsupset_P \{\!|b|\!\}$ and $\{\!|c|\!\} \sqsupset_Q \{\!|a|\!\} \sqsupset_Q \{\!|\,|\!\} \sqsupset_Q \{\!|d|\!\}$. Then:

- If the environment offers $\{\!|a|\!\}$ then P will perform it;
- If the environment does not offer $\{\!|a|\!\}$, then P may idle and Q may perform $\{\!|c|\!\}$, $\{\!|\,|\!\}$ or $\{\!|d|\!\}$. Note that even if the environment doesn't allow idling at some time t (for example if it offers only $\{\!|c|\!\}$ or $\{\!|d|\!\}$) then P may idle at time t while Q performs $\{\!|c|\!\}$ or $\{\!|d|\!\}$. Note also that Q cannot perform $\{\!|a|\!\}$ since if the environment offers it, it would be performed by P.
- If none of these are possible, then P will perform $\{\!|b|\!\}$.

Hence $P \boxempty Q$ has an offer relation with $\{\!|a|\!\} \sqsupset \{\!|c|\!\} \sqsupset \{\!|\,|\!\} \sqsupset \{\!|d|\!\} \sqsupset \{\!|b|\!\}$.
In general, the offer relation of $P \boxempty Q$ is formed by

1. taking P's offer relation ($\{\!|a|\!\} \sqsupset \{\!|\,|\!\} \sqsupset \{\!|b|\!\}$ in our example);
2. replacing the occurrence of $\{\!|\,|\!\}$ with Q's offer relation (to get $\{\!|a|\!\} \sqsupset \{\!|c|\!\} \sqsupset \{\!|a|\!\} \sqsupset \{\!|\,|\!\} \sqsupset \{\!|d|\!\} \sqsupset \{\!|b|\!\}$ in our example);
3. for each bag that occurs twice, removing the lower copy (to get $\{\!|a|\!\} \sqsupset \{\!|c|\!\} \sqsupset \{\!|\,|\!\} \sqsupset \{\!|d|\!\} \sqsupset \{\!|b|\!\}$).

In general, $P \boxempty Q$ can perform the offer w if the environment offers w and

- P would rather perform w than idle and the environment offers nothing that P prefers to w;
- P chooses to idle, Q offers w and the environment offers nothing that Q prefers to w; *or*
- Q doesn't offer w, P would rather idle than perform w but the environment does not allow idling and does not offer anything that Q could perform nor anything that P prefers to w.

The process should offer w more strongly than v if

- P prefers w to idling and v is either offered by Q but not P, or offered by P less strongly than w;
- P prefers neither v nor w to idling, Q offers w and either Q prefers w to v, *or* v is offered by P but not Q; *or*
- P offers v weaker than w, but would rather idle, and Q offers neither v or w.

Hence if P has offer relation $\sqsubseteq_P$ and Q has offer relation $\sqsubseteq_Q$ then $P \boxempty Q$ has offer relation $\sqsubseteq_P \boxempty \sqsubseteq_Q$, where the operator $\_\boxempty\_ : \mathit{OFFREL} \times \mathit{OFFREL} \to \mathit{OFFREL}$ is defined by

$$\begin{aligned}
v(\sqsubseteq_P \boxempty \sqsubseteq_Q)w \;\Leftrightarrow\; & w \sqsupset_P (t, \{\!|\,|\!\}) \wedge (v \sqsubseteq_P w \vee v \in \mathrm{items} \sqsubseteq_Q \setminus \mathrm{items} \sqsubseteq_P) \\
& \vee\, v, w \not\sqsupset_P (t, \{\!|\,|\!\}) \wedge w \in \mathrm{items} \sqsubseteq_Q \wedge (v \sqsubseteq_Q w \vee v \in \mathrm{items} \sqsubseteq_P \setminus \mathrm{items} \sqsubseteq_Q) \\
& \vee\, v \sqsubseteq_P w \sqsubset_P (t, \{\!|\,|\!\}) \wedge v, w \notin \mathrm{items} \sqsubseteq_Q \\
& \quad \text{where } t = Iv = Iw
\end{aligned}$$

Note that $\mathrm{items}(\sqsubseteq_P \boxempty \sqsubseteq_Q) = \mathrm{items} \sqsubseteq_P \cup \mathrm{items} \sqsubseteq_Q$.
Having explained how the offer relation of $P \boxempty Q$ is derived from the offer relations of P and Q, we can now derive the semantic definition of the process. The process $P \boxempty Q$ can

- perform the empty trace if both P and Q can;

- perform a non-empty trace s if P can perform it and Q can perform the empty trace up until time $t = \mathit{begin}\, s$; if the bag performed at time t is below the empty bag in P's offer relation, then Q must also be able to reject it (or else Q would have performed it); *or*
- perform a non-empty trace s if Q can perform it and P can perform the empty trace up until time $\mathit{begin}\, s$; P must not prefer the initial action to idling (or else P would have performed it).

This gives the following definition:

$$
\begin{aligned}
&\mathcal{A}_{DT}[\![P \mathbin{[\!\!\Box} Q]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, \sqsubseteq_P \mathbin{[\!\!\Box} \sqsubseteq_Q, \prec\succ) \mid (\tau, \sqsubseteq_P, \prec\succ) \in A_P \wedge (\tau, \sqsubseteq_Q, \prec\succ) \in A_Q\} \\
&\quad \cup \{(\tau, \sqsubseteq_P \mathbin{[\!\!\Box} \sqsubseteq_Q, s) \mid \\
&\qquad\quad s \neq \prec\succ \wedge \mathit{begin}\, s = t \wedge (t, \sqsubseteq_P, \prec\succ) \in A_P \wedge (\tau, \sqsubseteq_Q, s) \in A_Q \wedge s \uparrow t \not\sqsupset_P (t, \{\!|\,|\!\})\} \\
&\quad \cup \{(\tau, \sqsubseteq_P \mathbin{[\!\!\Box} \sqsubseteq_Q, s) \mid \\
&\qquad\quad s \neq \prec\succ \wedge \mathit{begin}\, s = t \wedge (\tau, \sqsubseteq_P, s) \in A_P \wedge (t, \sqsubseteq_Q, \prec\succ) \in A_Q \\
&\qquad\quad \wedge (s \uparrow t \sqsupset_P (t, \{\!|\,|\!\}) \vee s \uparrow t \notin \mathrm{items} \sqsubseteq_Q)\}
\end{aligned}
$$

We define $P \mathbin{\Box\!\!]} Q$ by $P \mathbin{\Box\!\!]} Q \mathrel{\widehat{=}} Q \mathbin{[\!\!\Box} P$.

We have a number of laws for the left-biased choice operator:

$$
\begin{array}{rcll}
(P \mathbin{[\!\!\Box} Q) \mathbin{[\!\!\Box} R & = & P \mathbin{[\!\!\Box} (Q \mathbin{[\!\!\Box} R) & \text{associativity} \\
P \mathbin{[\!\!\Box} P & = & P & \text{idempotence} \\
P \mathbin{[\!\!\Box} \mathit{STOP} = \mathit{STOP} \mathbin{[\!\!\Box} P & = & P & \mathit{STOP} \text{ is an identity}
\end{array}
$$

Similar laws hold for the right-biased choice operator.

The following 'laws' do not hold:

$$
P \mathbin{[\!\!\Box} (Q \mathbin{\Box\!\!]} R) = (P \mathbin{[\!\!\Box} Q) \mathbin{\Box\!\!]} R \qquad P \mathbin{\Box\!\!]} (Q \mathbin{[\!\!\Box} R) = (P \mathbin{\Box\!\!]} Q) \mathbin{[\!\!\Box} R
$$

Let $P \mathrel{\widehat{=}} a \longrightarrow \mathit{STOP}$, $Q \mathrel{\widehat{=}} b \longrightarrow \mathit{STOP}$, $R \mathrel{\widehat{=}} c \longrightarrow \mathit{STOP}$. Then $P \mathbin{[\!\!\Box} (Q \mathbin{\Box\!\!]} R)$ and $(P \mathbin{\Box\!\!]} Q) \mathbin{[\!\!\Box} R$ will perform an a in preference to a c, whereas $(P \mathbin{[\!\!\Box} Q) \mathbin{\Box\!\!]} R$ and $P \mathbin{\Box\!\!]} (Q \mathbin{[\!\!\Box} R)$ will perform a c in preference to an a.

5.6 Parallel composition

We consider now the parallel composition of two processes. We start by considering the left-biased parameterized parallel composition, $P \;{}^{X}\!\!\nparallel^{Y}\; Q$. The offer relation of $P \;{}^{X}\!\!\nparallel^{Y}\; Q$ is derived from the offer relations of P and Q. $P \;{}^{X}\!\!\nparallel^{Y}\; Q$ will offer a bag of events w if P offers $w \restriction X$, Q offers $w \restriction Y$, *and* all the events of w are in either X or Y. Informally, the priorities of $P \;{}^{X}\!\!\nparallel^{Y}\; Q$ follow the priorities of P on X; more formally, a bag of events w is offered more strongly than w' if

- P offers $w \restriction X$ more strongly than $w' \restriction X$; *or*
- $w \restriction X = w' \restriction X$ and Q offers $w \restriction Y$ more strongly than $w' \restriction Y$.

Hence if P has offer relation $\sqsubseteq_P$ and Q has offer relation $\sqsubseteq_Q$ then the offer relation is $\sqsubseteq_P \;{}^{X}\!\!\nparallel^{Y} \sqsubseteq_Q$, where the function $\_ \;{}^{X}\!\!\nparallel^{Y}\; \_ : \mathit{OFFREL} \times \mathit{OFFREL} \to \mathit{OFFREL}$ is defined by

$$
\begin{aligned}
&w(\sqsubseteq_P \;{}^{X}\!\!\nparallel^{Y} \sqsubseteq_Q)w' \Leftrightarrow \\
&\quad (w \restriction X \sqsubset_P w' \restriction X \vee w \restriction X = w' \restriction X \wedge w \restriction Y \sqsubseteq_Q w' \restriction Y) \\
&\quad \wedge w \restriction X, w' \restriction X \in \mathrm{items} \sqsubseteq_P \wedge w \restriction Y, w' \restriction Y \in \mathrm{items} \sqsubseteq_Q \wedge \Sigma w, \Sigma w' \subseteq X \cup Y
\end{aligned}
$$

Note that $\text{items}(\sqsubseteq_P {}^{X}\!\lhd\!\!\parallel^{Y} \sqsubseteq_Q) = \{w \mid w \upharpoonright X \in \text{items} \sqsubseteq_P \wedge w \upharpoonright Y \in \text{items} \sqsubseteq_Q \wedge \Sigma w \subseteq X \cup Y\}$.
$P \;{}^{X}\!\lhd\!\!\parallel^{Y}\; Q$ will perform trace s if the alphabet of s is contained in $X \cup Y$, P can perform $s \upharpoonright X$, *and* Q can perform $s \upharpoonright Y$. Hence we have the following definition for parallel composition:

$$\begin{array}{l}\mathcal{A}_{DT}[\![P \;{}^{X}\!\lhd\!\!\parallel^{Y}\; Q]\!] \mathrel{\widehat{=}} \\ \quad \{(\tau, \sqsubseteq_P {}^{X}\!\lhd\!\!\parallel^{Y} \sqsubseteq_Q, s) \mid (\tau, \sqsubseteq_P, s \upharpoonright X) \in A_P \wedge (\tau, \sqsubseteq_Q, s \upharpoonright Y) \in A_Q \wedge \Sigma s \subseteq X \cup Y\}\end{array}$$

We use this definition to define the other parallel operators:

$$P \;{}^{X}\!\parallel\!\!\rhd^{Y}\; Q \mathrel{\widehat{=}} Q \;{}^{Y}\!\lhd\!\!\parallel^{X}\; P \qquad P \lhd\!\!\parallel Q \mathrel{\widehat{=}} P \;{}^{\Sigma}\!\lhd\!\!\parallel^{\Sigma}\; Q \qquad P \parallel\!\!\rhd Q \mathrel{\widehat{=}} P \;{}^{\Sigma}\!\parallel\!\!\rhd^{\Sigma}\; Q$$

This gives the following:

$$\mathcal{A}_{DT}[\![P \lhd\!\!\parallel Q]\!] = \{(\tau, \sqsubseteq_P \lhd\!\!\parallel \sqsubseteq_Q, s) \mid (\tau, \sqsubseteq_P, s) \in A_P \wedge (\tau, \sqsubseteq_Q, s) \in A_Q\}$$

where the parallel composition of offers is defined by

$$w(\sqsubseteq_P \lhd\!\!\parallel \sqsubseteq_Q)w' \Leftrightarrow w \sqsubseteq_P w' \wedge w, w' \in \text{items} \sqsubseteq_Q$$

Note that $\text{items}(\sqsubseteq_P \lhd\!\!\parallel \sqsubseteq_Q) = \text{items} \sqsubseteq_P \cap \text{items} \sqsubseteq_Q$.
A number of laws hold for the parallel operators:

$$\begin{array}{rcll}
P \;{}^{X}\!\lhd\!\!\parallel^{Y \cup Z}\; (Q \;{}^{Y}\!\lhd\!\!\parallel^{Z}\; R) & = & (P \;{}^{X}\!\lhd\!\!\parallel^{Y}\; Q) \;{}^{X \cup Y}\!\lhd\!\!\parallel^{Z}\; R & \text{associativity} \\
P \parallel\!\!\rhd (Q \lhd\!\!\parallel R) & = & (P \parallel\!\!\rhd Q) \lhd\!\!\parallel R & \\
P \lhd\!\!\parallel (Q \lhd\!\!\parallel R) & = & P \lhd\!\!\parallel (Q \parallel\!\!\rhd R) & \\
P \lhd\!\!\parallel STOP = STOP \lhd\!\!\parallel P & = & STOP & STOP \text{ is a zero} \\
(a \longrightarrow P) \lhd\!\!\parallel (a \longrightarrow Q) & = & a \longrightarrow (P \lhd\!\!\parallel Q) & \text{communication} \\
(a \longrightarrow P) \lhd\!\!\parallel (b \longrightarrow Q) & = & STOP & \text{deadlock}
\end{array}$$

Similar laws hold for the right-biased parallel operator. Note that we do not have the following 'laws':

$$\begin{array}{rcl}
P \;{}^{X}\!\parallel\!\!\rhd^{Y \cup Z}\; (Q \;{}^{Y}\!\lhd\!\!\parallel^{Z}\; R) & = & (P \;{}^{X}\!\parallel\!\!\rhd^{Y}\; Q) \;{}^{X \cup Y}\!\lhd\!\!\parallel^{Z}\; R \\
P \;{}^{X}\!\lhd\!\!\parallel^{Y \cup Z}\; (Q \;{}^{Y}\!\lhd\!\!\parallel^{Z}\; R) & = & P \;{}^{X}\!\lhd\!\!\parallel^{Y \cup Z}\; (Q \;{}^{Y}\!\parallel\!\!\rhd^{Z}\; R)
\end{array}$$

Let $P \mathrel{\widehat{=}} a \longrightarrow STOP$, $Q \mathrel{\widehat{=}} b \longrightarrow STOP$, $R \mathrel{\widehat{=}} c \longrightarrow STOP$, $X \mathrel{\widehat{=}} \{a\}$, $Y \mathrel{\widehat{=}} \{b\}$, $Z \mathrel{\widehat{=}} \{c\}$. Then:

- $P \;{}^{X}\!\parallel\!\!\rhd^{Y \cup Z}\; (Q \;{}^{Y}\!\lhd\!\!\parallel^{Z}\; R)$ prefers a c to an a, whereas $(P \;{}^{X}\!\parallel\!\!\rhd^{Y}\; Q) \;{}^{X \cup Y}\!\lhd\!\!\parallel^{Z}\; R$ prefers an a to a c;
- $P \;{}^{X}\!\lhd\!\!\parallel^{Y \cup Z}\; (Q \;{}^{Y}\!\lhd\!\!\parallel^{Z}\; R)$ prefers a b to a c, whereas $P \;{}^{X}\!\lhd\!\!\parallel^{Y \cup Z}\; (Q \;{}^{Y}\!\parallel\!\!\rhd^{Z}\; R)$ prefers a c to a b.

We also do not have the 'law' $P \lhd\!\!\parallel (Q \parallel\!\!\rhd R) = (P \lhd\!\!\parallel Q) \parallel\!\!\rhd R$. Let $P \mathrel{\widehat{=}} a \,\square\, b$, $Q \mathrel{\widehat{=}} a \,\square\, b$, $R \mathrel{\widehat{=}} a \,\square\, b$. Then $P \lhd\!\!\parallel (Q \parallel\!\!\rhd R)$ prefers an a to a b, whereas $(P \lhd\!\!\parallel Q) \parallel\!\!\rhd R$ prefers a b to an a.

5.7 Interleaving

We want to derive a definition for the offer relation of $P \lhd\!\!|\!|\!| \, Q$ in terms of the offer relations of P and Q. We begin by considering the question

> If $P \lhd\!\!|\!|\!| \, Q$ offers w, then what do P and Q offer?

It is clear that P must offer some suboffer of w, and Q must offer the rest of w. Let w_P be the suboffer of w that P offers strongest subject to the condition that Q can perform the rest of w. Let w_Q be the rest of w. We claim that $P \lhd\!\!|\!|\!| \, Q$ offering w corresponds to P offering w_P and Q offering w_Q, so P performs the suboffer of w that it prefers (so long as Q can perform the rest of w). We define an operator $\Uparrow_{\sqsubseteq_P,\sqsubseteq_Q}$ which returns the subset of its argument that is offered strongest by $\sqsubseteq_P$ subject to the condition that the rest of the argument is offered by $\sqsubseteq_Q$.

$$\begin{aligned}&\Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} w = \sqcup_{\sqsubseteq_P}\{w'_P \in \text{items}\sqsubseteq_P \mid w'_P \subseteq w \wedge w - w'_P \in \text{items}\sqsubseteq_Q\}\\ &\quad \text{if } \exists\, w_P \in \text{items}\sqsubseteq_P, w_Q \in \text{items}\sqsubseteq_Q \bullet w = w_P \uplus w_Q\end{aligned}$$

Let w_P and w_Q be the suboffers of w performed by P and Q respectively. Let v_P and v_Q be the corresponding suboffers of v. In the combination $P \overleftarrow{|||} Q$ the process P is the master, so $P \overleftarrow{|||} Q$ offers w more strongly than v if P offers w_P strictly stronger than v_P, *or* $w_P = v_P$ and Q offers w_Q stronger than v_Q. Hence, if P and Q have offer relations $\sqsubseteq_P$ and $\sqsubseteq_Q$, the offer relation for $P \overleftarrow{|||} Q$ is $\sqsubseteq_P \overleftarrow{|||} \sqsubseteq_Q$ where the operator $\overleftarrow{|||}$ is defined on offer relations by

$$\begin{aligned}v(\sqsubseteq_P \overleftarrow{|||} \sqsubseteq_Q)w \;\Leftrightarrow\;& \exists\, v'_P \in \text{items}\sqsubseteq_P, v'_Q \in \text{items}\sqsubseteq_Q \bullet v = v'_P \uplus v'_Q\\ & \wedge \exists\, w'_P \in \text{items}\sqsubseteq_P, w'_Q \in \text{items}\sqsubseteq_Q \bullet w = w'_P \uplus w'_Q\\ & \wedge v_P \sqsubset_P w_P \vee v_P = w_P \wedge v_Q \sqsubseteq_Q w_Q\\ & \text{where } \; v_P = \Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} v \qquad v_Q = v - v_P\\ & \qquad\quad\; w_P = \Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} w \qquad w_Q = w - w_P\end{aligned}$$

Note that $\text{items}(\sqsubseteq_P \overleftarrow{|||} \sqsubseteq_Q) = \{w_P \uplus w_Q \mid w_P \in \text{items}\sqsubseteq_P \wedge w_Q \in \text{items}\sqsubseteq_Q\}$.
$P \overleftarrow{|||} Q$ can perform trace s if at all times t, P can perform some subbag of $s \uparrow t$ and Q can perform the rest of $s \uparrow t$. In particular, P performs the subbag of $s \uparrow t$ that it offers strongest subject to the condition that Q can perform the rest of $s \uparrow t$. We extend the $\Uparrow_{\sqsubseteq_P,\sqsubseteq_Q}$ operator to traces:

$$\Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} s \mathrel{\widehat=} \{t \mapsto \Uparrow_{\sqsubseteq_P,\sqsubseteq_Q}(s \uparrow t) \mid t \in Is\}$$

This gives the following definition:

$$\mathcal{A}_{DT}[\![P \overleftarrow{|||} Q]\!] \mathrel{\widehat=} \{(\tau, \sqsubseteq_P \overleftarrow{|||} \sqsubseteq_Q, s)) \mid (\tau, \sqsubseteq_P, \Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} s) \in A_P \wedge (\tau, \sqsubseteq_Q, s - \Uparrow_{\sqsubseteq_P,\sqsubseteq_Q} s) \in A_Q\}$$

The right biased interleave operator is defined by $P \overrightarrow{|||} Q \mathrel{\widehat=} Q \overleftarrow{|||} P$.
We have a number of laws for interleaving:

$$\begin{aligned}P \overleftarrow{|||} (Q \overleftarrow{|||} R) &= (P \overleftarrow{|||} Q) \overleftarrow{|||} R && \text{associativity}\\ P \overleftarrow{|||} \mathit{STOP} = \mathit{STOP} \overleftarrow{|||} P &= P && \mathit{STOP} \text{ is an identity}\end{aligned}$$

And similar laws hold for the right-biased interleaving. Note that we do not have the following 'laws':

$$P \overleftarrow{|||} (Q \overrightarrow{|||} R) = (P \overleftarrow{|||} Q) \overrightarrow{|||} R \qquad P \overrightarrow{|||} (Q \overleftarrow{|||} R) = (P \overrightarrow{|||} Q) \overleftarrow{|||} R$$

Let $P \mathrel{\widehat=} a \longrightarrow \mathit{STOP}$, $Q \mathrel{\widehat=} b \longrightarrow \mathit{STOP}$, $R \mathrel{\widehat=} c \longrightarrow \mathit{STOP}$. Then $P \overleftarrow{|||} (Q \overrightarrow{|||} R)$ and $(P \overrightarrow{|||} Q) \overleftarrow{|||} R$ prefer a to c whereas $(P \overleftarrow{|||} Q) \overrightarrow{|||} R$ and $P \overrightarrow{|||} (Q \overleftarrow{|||} R)$ prefer c to a.

5.8 Hiding

In order to define the operation of hiding on processes we must first define hiding on offer relations. A bag of events w being offered by $P \setminus X$ corresponds to P offering a bag of events w' such that $w' \setminus X = w$. In general, P may be able to perform several bags w' such that $w' \setminus X = w$. We make the assumption that it performs the one that is maximal with respect to its offer relation — this can be thought of as a sort of maximal progress assumption.

We therefore want to define an operator $\Uparrow_{\sqsubseteq}^{g} : \mathit{OFF} \nrightarrow \mathit{OFF}$ (where g is a function of type $\mathrm{bag}\,\Sigma \to \mathrm{bag}\,\Sigma$, such as the hiding operator on bags) such that $\Uparrow_{\sqsubseteq}^{g} w$ is the $\sqsubseteq$-strongest offer w' such that $gw' = w$.

$$\Uparrow_{\sqsubseteq}^{g} w = \sqcup_{\sqsubseteq}\{w' \in \mathrm{items} \sqsubseteq \mid gw' = w\} \qquad \text{if } \exists\, w' \in \mathrm{items} \sqsubseteq \bullet\, gw' = w$$

Hence our maximal progress assumption states that w being offered by $P \setminus X$ corresponds to $\Uparrow_{\sqsubseteq}^{-\setminus X} w$ being offered by P. The operator $\Uparrow_{\sqsubseteq}^{-\setminus X}$ can be though of as a sort of "inverse hiding" operator in the sense that $\_ \setminus X \circ \Uparrow_{\sqsubseteq}^{-\setminus X} = id$; the offer $\Uparrow_{\sqsubseteq}^{-\setminus X} w$ is the $\sqsubseteq$-maximal member of $(\_ \setminus X)^{-1}(w)$.

$P \setminus X$ will prefer w to v if P prefers $\Uparrow_{\sqsubseteq}^{-\setminus X} w$ to $\Uparrow_{\sqsubseteq}^{-\setminus X} v$. Hence we have the following definition:

$$v(\sqsubseteq \setminus X)w \;\Leftrightarrow\; \exists\, v', w' \in \mathrm{items} \sqsubseteq \bullet\, v' \setminus X = v \wedge w' \setminus X = w \wedge \Uparrow_{\sqsubseteq}^{-\setminus X} v \sqsubseteq \Uparrow_{\sqsubseteq}^{-\setminus X} w$$

Note that $\mathrm{items}(\sqsubseteq \setminus X) = \{v \setminus X \mid v \in \mathrm{items} \sqsubseteq\}$.

An offer relation $\sqsubseteq$ of $P \setminus X$ must have resulted from an offer relation $\sqsubseteq'$ of P, such that $\sqsubseteq' \setminus X = \sqsubseteq$. Then for $P \setminus X$ to perform trace s, P must perform trace $\Uparrow_{\sqsubseteq'}^{-\setminus X} s$ where the $\Uparrow_{-}^{-}$ operator is defined on traces by $\Uparrow_{\sqsubseteq'}^{g} s \mathrel{\widehat{=}} \{t \mapsto \Uparrow_{\sqsubseteq'}^{g}(s \uparrow t) \mid t \in Is\}$. This only exists if for all t there is some $v \in \mathrm{items} \sqsubseteq'$ such that $v \setminus X = s \uparrow t$, which is equivalent to saying $\forall\, t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq$. This gives the following definition:

$$\mathcal{A}_{DT}[\![P \setminus X]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \forall\, t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq \wedge \exists \sqsubseteq' \bullet \sqsubseteq' \setminus X = \sqsubseteq \wedge (\tau, \sqsubseteq', \Uparrow_{\sqsubseteq'}^{-\setminus X} s) \in A_P\}$$

The following laws relate to the hiding operator:

$$\begin{array}{rcll} P \setminus \{\} & = & P & \mathit{STOP} \text{ is an identity} \\ (P \setminus X) \setminus Y & = & P \setminus (X \cup Y) & \text{successive hiding} \\ (a \xrightarrow{t} P) \setminus X & = & \left\{ \begin{array}{ll} \mathit{WAIT}\ t\,;(P \setminus X) & \text{if } a \in X \\ a \xrightarrow{t} (P \setminus X) & \text{if } a \notin X \end{array} \right. & \text{distribution through prefixing} \end{array}$$

Also hiding distributes through parallel composition if the hidden events are a subset of the symmetric difference of the alphabets: if $X \subseteq A \setminus B \cup B \setminus A$ then

$$(P \;{}_{A}\!\|_{B}\; Q) \setminus X = (P \setminus X) \;{}_{A}\!\|_{B}\; (Q \setminus X)$$

5.9 Renaming

The definition of renaming is in many ways very similar to the definition of hiding. To define renaming on processes, we must first define renaming on offer relations. The process gP performing v corresponds to P performing $\Uparrow_{\sqsubseteq}^{g} v$ (assuming of course that there is some $v' \in \mathrm{items} \sqsubseteq$ such that $gv' = v$). Hence the offer relation renaming operator $\odot$ has the following definition:

$$v(g \odot \sqsubseteq)w \;\Leftrightarrow\; \exists\, v', w' \in \mathrm{items} \sqsubseteq \bullet\, gv' = v \wedge gw' = w \wedge \Uparrow_{\sqsubseteq}^{g} v \sqsubseteq \Uparrow_{\sqsubseteq}^{g} w$$

Note that $\mathrm{items}(g \odot \sqsubseteq) = \{gv \mid v \in \mathrm{items} \sqsubseteq\}$. We shall sometimes choose to write $g \odot \sqsubseteq$ as $g\sqsubseteq$.

A behaviour $(\tau, \sqsubseteq, s)$ of gP must correspond to a behaviour $(\tau, \sqsubseteq', \Uparrow_{\sqsubseteq'}^{g} s)$ of P, such that $g \odot \sqsubseteq' = \sqsubseteq$. This is well defined only if $\forall\, t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq$. This gives the following definitions:

$$\mathcal{A}_{DT}[\![gP]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \forall\, t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq \wedge \exists \sqsubseteq' \bullet g \odot \sqsubseteq' = \sqsubseteq \wedge (\tau, \sqsubseteq', \Uparrow_{\sqsubseteq'}^{g} s) \in A_P\}$$

For bijective g we have

$$\mathcal{A}_{DT}[\![gP]\!] \mathrel{\widehat{=}} \{(\tau, g \odot \sqsubseteq, gs) \mid (\tau, \sqsubseteq, s) \in A_P\}$$

Where in this case $g \odot \sqsubseteq = \sqsubseteq'$ where $gv \sqsubseteq' gw \Leftrightarrow v \sqsubseteq w$.

The following laws hold for the renaming operator:

$$\begin{array}{rcll} f(gP) & = & (f \circ g)P & \text{Successive renaming} \\ g(a \longrightarrow P) & = & ga \longrightarrow gP & \text{distribution through prefixing} \end{array}$$

The following distribution laws hold if g is a bijection:

$$\begin{array}{rcl} g(P\ {}^{X}\|^{Y}\ Q) & = & gP\ {}^{gX}\|^{gY}\ gQ \\ g(\mu\, X \bullet FX) & = & \mu\, Y \bullet g(F(g^{-1}\, Y)) \\ g(P \setminus X) & = & gP \setminus gX \\ g(P \mathbin{\Box} Q) & = & gP \mathbin{\Box} gQ \\ g(P \interleave Q) & = & gP \interleave gQ \end{array}$$

5.10 Sequential composition

$P \fatsemi Q$ first executes P; if P terminates then Q is executed after a delay of δ. A behaviour $(\tau, \sqsubseteq, s)$ of $P \fatsemi Q$ can come about in three ways:

- a behaviour of P that does not terminate before time τ;
- a behaviour of P that terminates between times $\tau - \delta$ and τ; *or*
- a behaviour of P that terminates successfully before time $\tau - \delta$ followed by a behaviour of Q after a delay of δ.

Note that we have to hide the event $\checkmark$ from any behaviour of P in order to make sure that it happens (silently) as soon as possible. We have the following definition:

$$\begin{aligned} \mathcal{A}_{DT}[\![P \fatsemi Q]\!] \mathrel{\widehat{=}} \{&(\tau, \sqsubseteq_P, s_P) \mid \forall\, t \bullet s_P \uparrow t \in \text{items} \sqsubseteq_P \\ &\quad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (\tau, \sqsubseteq'_P, \Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \checkmark \notin \Sigma(\Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P)\} \\ \cup\{&(\tau, \sqsubseteq_P{}^\frown(t,\tau] \otimes \langle \{\!|\!\}\rangle, s_P) \mid \\ &\quad t \leqslant \tau < t + \delta \wedge \forall\, t' \bullet s_P \uparrow t' \in \text{items} \sqsubseteq_P \\ &\quad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (t, \sqsubseteq'_P, \Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \mathit{begin}((\Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P) \upharpoonright \checkmark) = t\} \\ \cup\{&(\tau, \sqsubseteq_P{}^\frown(t, t+\delta) \otimes \langle \{\!|\!\}\rangle^\frown \sqsubseteq_Q + t + \delta, s_P{}^\frown s_Q + t + \delta) \mid \\ &\quad t \leqslant \tau - \delta \wedge \forall\, t' \bullet s_P \uparrow t' \in \text{items} \sqsubseteq_P \\ &\quad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (t, \sqsubseteq'_P, \Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \mathit{begin}((\Uparrow^{-\setminus\checkmark}_{\sqsubseteq'_P} s_P) \upharpoonright \checkmark) = t \\ &\quad \wedge (\tau - (t + \delta), \sqsubseteq_Q, s_Q) \in A_Q\} \end{aligned}$$

We have the following laws for sequential composition:

$$\begin{array}{rcll} (P \fatsemi Q) \fatsemi R & = & P \fatsemi (Q \fatsemi R) & \text{associativity} \\ (a \longrightarrow P) \fatsemi Q & = & a \longrightarrow (P \fatsemi Q) & \text{associativity with prefixing} \\ \mathit{STOP} \fatsemi P & = & \mathit{STOP} & \mathit{STOP} \text{ is a left zero} \end{array}$$

5.11 Delay

Consider a behaviour $(\tau, \sqsubseteq, s)$ of $\mathit{WAIT}\ t\, ;P$:

- if $t > \tau$ then the process can perform and offer nothing;
- if $t \leqslant \tau$ then the process can act like P, temporally shifted by t.

This gives the following definition:

$$\begin{aligned}\mathcal{A}_{DT}[\![WAIT\ t\ ; P]\!] \mathrel{\widehat=} & \{(\tau, [0, \tau] \otimes \langle \{\!|\!\}\rangle, \{\!|\!\}) \mid t > \tau\} \\ & \cup\{(\tau, \sqsubseteq + t, s + t) \mid t \leqslant \tau \wedge (\tau - t, \sqsubseteq, s) \in A_P\}\end{aligned}$$

The following laws hold:

$$\begin{array}{rcll} SKIP \,{}_{9}^{\circ}\, P &=& WAIT\ \delta\ ; P & \text{effect of } SKIP \\ WAIT\ t\ ; WAIT\ t'\ ; P &=& WAIT\ t + t'\ ; P & \text{successive delays}\end{array}$$

5.12 Recursion

In order to define recursion, we first define a metric on the space DTM_B. We do this by considering the first time at which two processes may be distinguished. We define an operator $\lhd$ on behaviour sets which gives the behaviours of a process up to a certain time:

$$A \lhd t \mathrel{\widehat=} \{(\tau, \sqsubseteq, s) \in A \mid \tau < t\}$$

We define the metric on DTM_B by

$$d(A_P, A_Q) \mathrel{\widehat=} \inf\{2^{-t} \mid A_P \lhd t = A_Q \lhd t\}$$

We want recursive calls to be delayed by time δ so as to make all recursions well defined. We define a function M_δ on the semantic space which delays all behaviours by time δ.

$$M_\delta A \mathrel{\widehat=} \{(\tau, [0, \tau] \otimes \langle \{\!|\!\}\rangle, \{\!|\!\}) \mid \tau < \delta\} \cup \{(\tau + \delta, [0, \delta) \otimes \langle \{\!|\!\}\rangle ^\frown \sqsubseteq + \delta, s + \delta) \mid (\tau, \sqsubseteq, s) \in A\}$$

We can now define recursion by

$$\begin{aligned}\mathcal{A}_{DT}[\![\mu\, X \bullet F(X)]\!] \mathrel{\widehat=} & \text{ the unique fixed point of the contraction mapping } C \circ M_\delta \\ & \text{ where } C \text{ is the mapping on } DTM_B \text{ represented by } F\end{aligned}$$

Part II
The Probabilistic Model

6 Syntax for the probabilistic model

We now start to discuss the probabilistic model. We add a probabilistic choice operator to the syntax of the language discussed in the previous sections: the process $P\ {}_p\sqcap_q\ Q$ will act like P with probability p and like Q with probability q. Since our previous language was completely deterministic, we know that the only place where nondeterminism arises in our new language is through the use of the probabilistic choice operator; this will allow us to give probabilities to all nondeterministic choices.

The complete syntax for Probabilistic Biased Timed CSP ($PBTCSP$) is as follows:

$$\begin{aligned} P ::= {} & STOP \mid SKIP \mid WAIT\ t \mid a \xrightarrow{t} P \mid P \mathrel{\boxminus} P \mid P \mathrel{\boxminus} P \mid P \mathrel{\#} P \mid P \mathrel{\#} P \mid \\ & P\ {}_A\!\#_B\ P \mid P\ {}_A\!\#_B\ P \mid P \leftarrow\!\!\#\ P \mid P\ \#\!\!\rightarrow P \mid P\ {}_p\sqcap_q\ P \mid P \,{}_{9}^{\circ}\, P \mid WAIT\ t\ ; P \mid \\ & P \setminus A \mid f(P) \mid \mu\, X \bullet F(X)\end{aligned}$$

where t ranges over the set $TIME$ of times; a ranges over some alphabet Σ of events; p and q range over the interval $(0, 1)$, with the property that $p + q = 1$; A and B range over $\mathbb{P}\,\Sigma$; f ranges over $\Sigma \to \Sigma$; and F ranges over functions from process names to processes.

7 Semantics

As before, we define a behaviour or an observation of a process to be a triple $(\tau, \sqsubseteq, s)$, where

- τ is the time up until which the process is observed.
- $\sqsubseteq$ is a partial order on the space OFF ($= TIME \times \text{bag}\,\Sigma$) of offers. We say a process offers χ stronger than ψ at time t, and write $(t, \psi) \sqsubseteq (t, \chi)$ if the process gives a higher priority to the bag of events χ than the bag of events ψ at time t.
- s is a timed trace, of type $TIME \twoheadrightarrow \text{bag}\,\Sigma$; $s(t)$ is the bag of events performed at time t.

Recall the definition of the space TS_B of sets of prioritized behaviours:

$$TS_B \mathrel{\widehat{=}} \mathbb{P}(BEH)$$

We also want to be able to discuss the space PTF_B (Probabilistic, Timed Functions using Biases) of probability functions:

$$PTF_B \mathrel{\widehat{=}} BEH \rightarrow [0, 1]$$

We will represent a process by a pair (A, f) of type $TS_B \times PTF_B$. As before, A gives the set of behaviours that a process can perform; f is a probability function: $f(\tau, \sqsubseteq, s)$ is the probability of $(\tau, \sqsubseteq, s)$ occurring, given a suitable environment, i.e. any environment Ω such that $(\tau, \sqsubseteq, s)$ is compatible with Ω (in the sense of section 4.4). Note that if $(\tau, \sqsubseteq, s)$ is compatible with two different environments Ω and Ω' then the probability of $(\tau, \sqsubseteq, s)$ occurring is the same in environment Ω as in Ω'. This is because to say that $(\tau, \sqsubseteq, s)$ is compatible with Ω and Ω' means that both environments offer everything performed in trace s but neither offers anything offered stronger under the offer relation $\sqsubseteq$: the rest of the environmental offers do not have any affect on the behaviour of the process so the probability of $(\tau, \sqsubseteq, s)$ is the same in each environment.

We define the space PTM_B (the Probabilistic Timed Model using Biases) to be those pairs (A, f) in $TS_B \times PTF_B$ satisfying the following axioms:

1. $\forall \tau \geqslant 0 \bullet \exists\, n(\tau) \bullet (\tau, \sqsubseteq, s) \in A \Rightarrow \# s \leqslant n(\tau)$
2. $\forall \tau \geqslant 0 \bullet \exists\, n(\tau) \bullet (\tau, \sqsubseteq, s) \in A \Rightarrow \exists\, k \leqslant n(\tau)\,;\, I_0, \ldots, I_{k-1} \in TINT \bullet$
   $I_0, \ldots I_{k-1}$ partition $[0, \tau]$
   $\wedge\, \forall\, i : 0 \,..\, k-1\,;\, t, t' \in I_i\,;\, \chi, \psi \in \text{bag}\,\Sigma \bullet (t, \chi) \sqsubseteq (t, \psi) \Leftrightarrow (t', \chi) \sqsubseteq (t', \psi)$
3. $(\tau, \sqsubseteq, s) \in A \wedge (t, \chi) \in \text{items} \sqsubseteq\, \Rightarrow (t, \sqsubseteq \trianglelefteq t, s \lhd t ^\frown (t, \chi)) \in A$
4. $f(\tau, \sqsubseteq, s) > 0 \Rightarrow (\tau, \sqsubseteq, s) \in A$
5. $\sum \{\!| f(0, \sqsubseteq, \prec\succ) \mid\, \sqsubseteq\, \in OFFREL |\!\} = 1$
6. $\forall\, s : TT\,;\, \sqsubseteq : OFFREL\,;\, \Omega : EOFF\,;\, \tau, \tau' : TIME \mid \text{dom}\, s = [0, \tau] \wedge I\Omega = [\tau, \tau'] \bullet$
   $f(\tau, \sqsubseteq, s) = \sum \{\!| f(\tau', \sqsubseteq', s \lhd \tau ^\frown \sqcup_{\sqsubseteq'} \Omega) \mid\, \sqsubseteq' \trianglelefteq \tau = \sqsubseteq |\!\}$

The first three of these axioms are the same as the first three axioms in the deterministic model. We discuss the other three axioms in turn:

4. If the probability of a process having a certain behaviour is non zero, then that behaviour is possible.
5. If the environment offers no events at time 0, then the empty trace occurs with probability one.

6. The probability of a process displaying some behaviour up to time τ is the same as the sum of the probabilities of the extensions of this behaviour that could have resulted from the environment offering Ω between times τ and τ'.

It is worth noting that, in any environment, there is a countable number of behaviours that a process can perform — this is a result of the syntax we have chosen, which only allows *binary* probabilistic choice. This fact means that summing over probabilities (rather than integrating) is a valid technique.

7.1 The rôle of the environment

It is worth stressing the relationship between the probability function f and the environment Ω. $f(\tau, \sqsubseteq, s)$ is the probability of the process performing $(\tau, \sqsubseteq, s)$ for behaviours $(\tau, \sqsubseteq, s)$ that are compatible with Ω. We can use this to define a probability function f_Ω (for each environment Ω) that gives the probabilities of each behaviour in environment Ω.

$$f_\Omega(\tau, \sqsubseteq, s) \mathrel{\widehat=} \begin{cases} f(\tau, \sqsubseteq, s) & \text{if } (\tau, \sqsubseteq, s) \mathit{\ compat\ } \Omega \\ 0 & \text{otherwise} \end{cases}$$

As an example, consider the process that models a biased coin being tossed once:

$$COIN \mathrel{\widehat=} head \longrightarrow STOP \;{}_{1/3}\sqcap_{2/3}\; tail \longrightarrow STOP$$

Here is a list of some of the possible behaviours of $COIN$ when it is observed up until time 2:

$$\begin{aligned}
b_1 &\mathrel{\widehat=} (2, && [0,2] \otimes \langle \{\!|head|\!\}, \{\!|\,|\!\}\rangle &&, \langle\rangle \;)\\
b_2 &\mathrel{\widehat=} (2, && [0,2] \otimes \langle \{\!|tail|\!\}, \{\!|\,|\!\}\rangle &&, \langle\rangle \;)\\
b_3 &\mathrel{\widehat=} (2, && [0,1] \otimes \langle \{\!|head|\!\}, \{\!|\,|\!\}\rangle ^\frown (1,2] \otimes \langle \{\!|\,|\!\}\rangle &&, \langle (1, head)\rangle)\\
b_4 &\mathrel{\widehat=} (2, && [0,1] \otimes \langle \{\!|tail|\!\}, \{\!|\,|\!\}\rangle ^\frown (1,2] \otimes \langle \{\!|\,|\!\}\rangle &&, \langle (1, tail)\rangle)
\end{aligned}$$

In behaviour b_1 the probabilistic choice is made in favour of the *head*, so a *head* is offered, but nothing is performed. Behaviour b_2 is similar, except the choice is made in favour of the *tail*. In behaviour b_3 the choice is made in favour of the *head*, which is performed at time 1. In behaviour b_4 the choice is made in favour of the *tail*, which is performed at time 1. The probability function f associated with this process associates the following probabilities with these behaviours:

$$f(b_1) = 1/3 \qquad f(b_2) = 2/3 \qquad f(b_3) = 1/3 \qquad f(b_4) = 2/3$$

The two behaviours where the probabilistic choice is made in favour of the *head* are given probability $1/3$ while the behaviours where the choice is made in favour of the *tail* are given probability $2/3$.

Consider now an environment Ω with duration $[0, 2]$ where neither a *head* nor a *tail* is offered. The behaviours b_1 and b_2 are compatible with this environment, but behaviours b_3 and b_4 are not, since in both of these an event is performed that was not offered by the environment — in fact b_1 and b_2 are the *only* behaviours that $COIN$ can perform in this environment. The probability function associated with this environment has

$$f_\Omega(b_1) = 1/3 \qquad f_\Omega(b_2) = 2/3 \qquad f_\Omega(b_3) = 0 \qquad f_\Omega(b_4) = 0$$

and all other behaviours are given probability zero. Note that the sum of the probabilities of the two possible behaviours is one.

Consider now an environment Ω that first offers a head at time 1, and does not offer a tail. Now behaviours b_2 and b_3 are the possible behaviours. Behaviour b_1 is incompatible with Ω because at time 1 it offers a *head* stronger than the empty bag, but performs the empty bag despite the fact that the environment is willing to perform a *head* — it disobeys the rule that says that at each instant the process must perform the member of the environmental offer that it offers strongest (i.e. is maximal in the process's offer relation). The probability function associated with this environmental offer therefore has

$$f_\Omega(b_1) = 0 \qquad f_\Omega(b_2) = 2/3 \qquad f_\Omega(b_3) = 1/3 \qquad f_\Omega(b_4) = 0$$

Finally, consider an environment that offers a *head* and a *tail* at time 1, but offers neither earlier. In this case behaviours b_3 and b_4 are possible; the other two are incompatible with the environmental offer. Hence the probability function associated with this environmental offer has

$$f_\Omega(b_1) = 0 \qquad f_\Omega(b_2) = 0 \qquad f_\Omega(b_3) = 1/3 \qquad f_\Omega(b_4) = 2/3$$

Note that the choice of whether the process offers a *head* or a *tail* is made at time 0, before either is actually offered by the environment.

The following law holds in this model: if the environment offers Ω, then the sum of the probabilities of all possible behaviours is one.

Theorem 7.1: $\forall \Omega : EOFF \bullet \forall \tau \geqslant end\,\Omega \bullet \sum \{\!| f(end\,\Omega, \sqsubseteq, \sqcup_{\sqsubseteq}\Omega) \mid \sqsubseteq \in OFFREL |\!\} = 1.$ ♡

7.2 Semantic functions

We shall define functions $\mathcal{A}_{PBT} : PBTCSP \to TS_B$ and $\mathcal{P}_{PBT} : PBTCSP \to PTF_B$ such that $\mathcal{A}_{PBT}[\![P]\!]$ gives the set of possible behaviour of process P, and $\mathcal{P}_{PBT}[\![P]\!]$ gives the behaviour probability function. We define the semantic function $\mathcal{F}_{PBT} : PBTCSP \to PTM_B$ by $\mathcal{F}_{PBT}[\![P]\!] \mathrel{\widehat=} (\mathcal{A}_{PBT}[\![P]\!], \mathcal{P}_{PBT}[\![P]\!])$. In the following section we give definitions for $\mathcal{A}_{PBT}$ and $\mathcal{P}_{PBT}$ for each of the constructs of the language. The definitions were proved sound in [7]. In section 8.6 we discuss which algebraic laws hold in this model.

8 Definitions of the operators

In this section we derive the semantic definitions for each of our basic processes and for each of our operators. For most of the processes (all except probabilistic choice and recursion) the definition of the set A of possible behaviours is the same as in the deterministic model; for these processes we derive the definitions for the probability functions from the definition of A. For probabilistic choice, the definitions are easy; for recursion, the definition is very similar to that in the deterministic model. The definitions are summarized in appendix A.

8.1 Basic processes

The processes $STOP$, $SKIP$ and $WAIT\ t$ are completely deterministic and have definitions for A of the form

$$\mathcal{A}_{PBT}[\![P]\!] \mathrel{\widehat=} \{(\tau, \sqsubseteq, s) \mid S(\tau, \sqsubseteq, s)\}$$

for some predicate S. Behaviours of this form occur with probability one; all other behaviours have probability zero. This gives the following definition:

$$\mathcal{P}_{PBT}[\![P]\!] \mathrel{\widehat=} \text{fillout}\{(\tau, \sqsubseteq, s) \mapsto 1 \mid S(\tau, \sqsubseteq, s)\}$$

where the function fillout : $(BEH \nrightarrow [0,1]) \to (BEH \to [0,1])$ extends partial behaviour probability functions to total probability functions:

$$\forall f, (\tau, \sqsubseteq, s) \bullet \text{fillout } f\ (\tau, \sqsubseteq, s) \mathrel{\widehat=} \begin{cases} f(\tau, \sqsubseteq, s) & \text{if } (\tau, \sqsubseteq, s) \in \text{dom}\, f \\ 0 & \text{if } (\tau, \sqsubseteq, s) \notin \text{dom}\, f \end{cases}$$

All behaviours not defined in f are assumed not to occur, and so are given zero probability.

8.2 Unary operators

Let F be one of the unary operators prefixing, hiding, renaming, or delay. For each of these operators the definition from section 5 can be put in the form

$$\mathcal{A}_{PBT}[\![F(P)]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \exists \tau', \sqsubseteq', s' \bullet (\tau', \sqsubseteq', s') \in \mathcal{A}_{PBT}[\![P]\!] \wedge S(\tau, \tau', \sqsubseteq, \sqsubseteq', s, s')\}$$

for some predicate S. $F(P)$ performing a behaviour $(\tau, \sqsubseteq, s)$ corresponds to P performing a behaviour $(\tau', \sqsubseteq', s')$ such that $S(\tau, \tau', \sqsubseteq, \sqsubseteq', s, s')$. Hence the probability of $F(P)$ performing the behaviour $(\tau, \sqsubseteq, s)$ is the probability of P performing some such behaviour $(\tau', \sqsubseteq', s')$; we want to sum over all such behaviours. This gives the following definition:

$$\mathcal{P}_{PBT}[\![F(P)]\!] \mathrel{\widehat{=}} \sum \{\!\!\{ \mathcal{P}_{PBT}[\![P]\!](\tau', \sqsubseteq', s') \mid S(\tau, \tau', \sqsubseteq, \sqsubseteq', s, s') \}\!\!\}$$

Note that this can normally be greatly simplified using the one-point rule.

8.3 Binary operators

Let $\oplus$ be one of the binary operators external choice, parallel composition, interleaving, or sequential composition. For each of these operators, the definition from section 5 can be put in the form

$$\begin{aligned}\mathcal{A}_{PBT}[\![P \oplus Q]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \exists\, &\tau_P, \tau_Q, \sqsubseteq_P, \sqsubseteq_Q, s_P, s_Q \bullet \\ &(\tau_P, \sqsubseteq_P, s_P) \in \mathcal{A}_{PBT}[\![P]\!] \wedge (\tau_Q, \sqsubseteq_Q, s_Q) \in \mathcal{A}_{PBT}[\![Q]\!] \\ &\wedge S(\tau, \tau_P, \tau_Q, \sqsubseteq, \sqsubseteq_P, \sqsubseteq_Q, s, s_P, s_Q)\}\end{aligned}$$

for some predicate S. The probability of $P \oplus Q$ performing such a behaviour $(\tau, \sqsubseteq, s)$ is the probability of P and Q performing some corresponding behaviours $(\tau_P, \sqsubseteq_P, s_P)$ and $(\tau_Q, \sqsubseteq_Q, s_Q)$ such that $S(\tau, \tau_P, \tau_Q, \sqsubseteq, \sqsubseteq_P, \sqsubseteq_Q, s, s_P, s_Q)$; hence we want to sum over all such behaviours. This gives the following definition:

$$\begin{aligned}&\mathcal{P}_{PBT}[\![P \oplus Q]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \\ &\quad \sum \{\!\!\{ \mathcal{P}_{PBT}[\![P]\!](\tau_P, \sqsubseteq_P, s_P).\mathcal{P}_{PBT}[\![Q]\!](\tau_Q, \sqsubseteq_Q, s_Q) \mid S(\tau, \tau_P, \tau_Q, \sqsubseteq, \sqsubseteq_P, \sqsubseteq_Q, s, s_P, s_Q) \}\!\!\}\end{aligned}$$

8.4 Recursion

In order to define recursion, we define a metric on the space PTM_B. We do this by considering the first time at which two processes may be distinguished. We define operators $\lhd$ on behaviour sets and behaviour probability functions that give the behaviour of a process up to a certain time:

$$A \lhd t \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \in A \mid \tau < t\} \qquad f \lhd t \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mapsto f(\tau, \sqsubseteq, s) \mid \tau < t\}$$

We define the metric on PTM_B by

$$d((A_P, f_P), (A_Q, f_Q)) \mathrel{\widehat{=}} \inf\{2^{-t} \mid A_P \lhd t = A_Q \lhd t \wedge f_P \lhd t = f_Q \lhd t\}$$

As in the prioritized model, we define a mapping M_δ that delays it argument by δ:

$$\begin{aligned}&M_\delta(A, f) \mathrel{\widehat{=}} (A', f') \\ &\quad \text{where } A' = \{(\tau, [0, \tau] \otimes \langle \{\!\!\}\!\!\} \rangle, \{\!\!\}\!\!\}) \mid \tau < \delta\} \cup \{(\tau + \delta, [0, \delta) \otimes \langle \{\!\!\}\!\!\} \rangle ^\frown \sqsubseteq + \delta, s + \delta) \mid (\tau, \sqsubseteq, s) \in A\} \\ &\qquad\quad f' = \text{fillout}\,(\{(\tau, [0, \tau] \otimes \langle \{\!\!\}\!\!\} \rangle, \{\!\!\}\!\!\}) \mapsto 1 \mid \tau < \delta\} \\ &\qquad\qquad \cup \{(\tau, \sqsubseteq, s) \mapsto f(\tau - \delta, \sqsubseteq - \delta, s - \delta) \mid \\ &\qquad\qquad\qquad \tau \geqslant \delta \wedge s \lhd \delta = \prec\succ \wedge \sqsubseteq \lhd \delta = [0, \delta) \otimes \langle \{\!\!\}\!\!\} \rangle\})\end{aligned}$$

We can now define recursion by

$$\begin{aligned}\mathcal{F}_{PBT}[\![\mu X \bullet F(X)]\!] \mathrel{\widehat{=}}\ &\text{the unique fixed point of the contraction mapping } C \circ M_\delta \\ &\text{where } C \text{ is the mapping on } PTM_B \text{ represented by } F\end{aligned}$$

8.5 Probabilistic choice

The process $P \,{}_p\sqcap_q\, Q$ acts like P with probability p, and like Q with probability q. It will have behaviour $(\tau, \sqsubseteq, s)$ if P is chosen and P has behaviour $(\tau, \sqsubseteq, s)$, *or* Q is chosen and Q has behaviour $(\tau, \sqsubseteq, s)$. We therefore have the following definitions, assuming $p \neq 0$, $q \neq 0$, and $p + q = 1$:

$$\begin{aligned} \mathcal{A}_{PBT}[\![P \,{}_p\sqcap_q\, Q]\!] &\mathrel{\widehat{=}} \mathcal{A}_{PBT}[\![P]\!] \cup \mathcal{A}_{PBT}[\![Q]\!] \\ \mathcal{P}_{PBT}[\![P \,{}_p\sqcap_q\, Q]\!](\tau, \sqsubseteq, s) &\mathrel{\widehat{=}} p.\mathcal{P}_{PBT}[\![P]\!](\tau, \sqsubseteq, s) + q.\mathcal{P}_{PBT}[\![Q]\!](\tau, \sqsubseteq, s) \end{aligned}$$

8.6 Algebraic laws

In this section we discuss which algebraic laws hold in our language. Some of these laws were proved in [6] and [7].

All the laws that were described above for the deterministic model carry forward to this model with the exception of the idempotence of the external choice operators. Consider the process $P \mathrel{\widehat{=}} a \,{}_p\sqcap_q\, b$; $P \,\square\, P$ can offer both a and b at the same time, but P cannot.

In addition, the probabilistic choice operator is commutative, idempotent and associative:

$$P \,{}_p\sqcap_q\, Q = Q \,{}_q\sqcap_p\, P \qquad P \,{}_p\sqcap_q\, P = P \qquad P \,{}_p\sqcap_{q+r}\, (Q \,{}_{q/q+r}\sqcap_{r/q+r}\, R) = (P \,{}_{p/p+q}\sqcap_{q/p+q}\, Q) \,{}_{p+q}\sqcap_r\, R$$

It also distributes through all the other operators except recursion.

9 Example: a communications protocol

We consider a very simple communications protocol, transmitting over an unreliable medium. For simplicity, we abstract away from the actual contents of the communication, and consider a protocol that handles only a single message. We are interested in the probability of the message being correctly transmitted within a certain time.

The protocol is as in figure 3. Messages are received on the channel *in*. They are then passed along the wire W, which loses a proportion of its inputs. If Q receives the message, it acknowledges it on the channel *ack* and outputs on *out*. If P does not receive an acknowledgement within a certain amount of time, then it passes the message along W again.

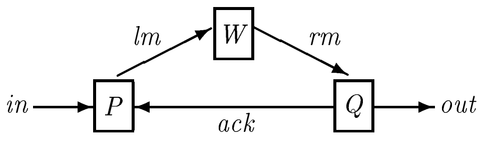

Figure 3: The communications protocol

The processes P, Q and W are defined by

$$\begin{aligned} P &\mathrel{\widehat{=}} in \longrightarrow \mu X \bullet lm \longrightarrow (ack \longrightarrow STOP \,\square\, WAIT\ 1 - 2\delta \,;\, X) \\ Q &\mathrel{\widehat{=}} rm \longrightarrow ack \xrightarrow{1-3\delta} out \longrightarrow STOP \\ W &\mathrel{\widehat{=}} \mu X \bullet lm \longrightarrow ((rm \longrightarrow X) \,{}_p\sqcap_q\, X) \end{aligned}$$

The protocol is then given by

$$PROTOCOL \mathrel{\widehat{=}} ((P \,{}^{A}\|^{B}\, W) \,{}^{A \cup B}\|^{C}\, Q) \setminus Y$$

where A, B and C are the alphabets of P, W and Q, and Y is the set of internal actions:

$$A \mathrel{\widehat{=}} \{in, lm, ack\} \qquad B \mathrel{\widehat{=}} \{lm, rm\} \qquad C \mathrel{\widehat{=}} \{rm, out, ack\} \qquad Y \mathrel{\widehat{=}} \{lm, rm, ack\}$$

For simplicity we rewrite P and W by

$$\begin{array}{rcl} P &=& in \longrightarrow P_1 \\ P_1 &=& lm \longrightarrow (ack \longrightarrow STOP \boxdot WAIT\ 1 - \delta\ ; P_1) \\ W &=& lm \longrightarrow (rm \xrightarrow{2\delta} W\ {}_p\sqcap_q\ WAIT\ \delta\ ;\ W) \end{array}$$

Then using laws for communication and hiding we have

$$PROTOCOL = in \longrightarrow PROTOCOL' \qquad \text{where } PROTOCOL' \mathrel{\widehat{=}} ((P_1\ {}^{A}\!\!\between^{B}\ W)\ {}^{A\cup B}\!\!\between^{C}\ Q) \setminus Y$$

Again using the algebraic laws presented in the previous sections, we can show

$$PROTOCOL' = WAIT\ 1 - \delta\ ;\ out\ {}_p\sqcap_q\ WAIT\ 1\ ;\ PROTOCOL'$$

Let q_n be the probability that $PROTOCOL'$ is *not* willing to perform *out* within $n - \delta$ seconds ($n \in \mathbb{N}$). Evidently $q_0 = 1$ and $q_{n+1} = q.q_n$. Hence $q_n = q^n$ and so the protocol *is* willing to perform *out* within n seconds of receiving an input with probability $1 - q^n$. Letting n tend to infinity we see that the protocol is eventually willing to perform *out* with probability one.

Part III

Summary

In this paper we have produced two languages with associated semantic models that are refinements of the timed failures model of timed CSP.

Through the use of biased operators we have defined a totally deterministic language. We have described a semantic model for the language which models the different priorities for actions, and have given semantic definitions for all the constructs of the language. We have illustrated our language and model with a couple of examples, implementing systems that would be very difficult without biased operators. We believe that this language gives us a more powerful framework for describing distributed systems.

We have then extended the language to include a probabilistic choice operator. This has allowed us to present a semantics which models the probabilities of different behaviours occurring. We have illustrated our model with an example. Our model will help in reasoning about stochastic behaviour of distributed systems. It will allow the specification and verification of fault tolerant processes, and the modelling of fairness.

In this paper we have only given a brief overview of the prioritized and probabilistic languages and their semantic models. In this last section we will give a brief summary of more recent work on this topic.

In [8] we added a number of new operators to the syntax, such as interrupt and infinite probabilistic choice operators. We also added variable bindings to our semantic model; this allowed recursion to be handled in a much neater way.

In [9] we presented a proof system for a language using prioritized operators. We write P **sat** $S(\tau, \sqsubseteq, s)$, where S is a predicate with free variable representing a behaviour, to mean that *all* behaviours of a process satisfy the predicate S:

$$P \textbf{ sat } S(\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \forall (\tau, \sqsubseteq, s) \in \mathcal{A}_{PBT}[\![P]\!] \bullet S(\tau, \sqsubseteq, s)$$

We described a language for writing specifications, based upon the language presented in [3]; the form of specifications written in this language is as near to the English language as possible, so that we can be reasonably confident that they capture our requirements. For example, we write $(a \ \mathsf{from}\ t)(\tau, \sqsubseteq, s)$ to specify that the process is willing to do an a at any time from t until an a is performed:

$$(a \ \mathsf{from}\ t)(\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \forall\, t' \in [t, \mathit{begin}(s \upharpoonright a \trianglerighteq t) \sqcap \tau] \bullet (t', \{\!|a|\!\}) \in \text{items} \sqsubseteq$$

We then presented a number of inference rules for our language. For composite processes, a proof obligation is reduced to proof obligations on the subcomponents; for example, the proof rule for left-biased parallel composition is

$$\frac{\begin{array}{l} P \ \mathbf{sat}\ S_P(\tau, \sqsubseteq, s) \\ Q \ \mathbf{sat}\ S_Q(\tau, \sqsubseteq, s) \\ S_P(\tau, \sqsubseteq_P, s) \wedge S_Q(\tau, \sqsubseteq_Q, s) \Rightarrow S(\tau, \sqsubseteq_P \Lleftarrow\!\!\Vert \sqsubseteq_Q, s) \end{array}}{P \Lleftarrow\!\!\Vert\ Q \ \mathbf{sat}\ S(\tau, \sqsubseteq, s)}$$

If P satisfies S_P and Q satisfies S_Q, and if a behaviour of $P \Lleftarrow\!\!\Vert\ Q$ satisfies S *whenever* the corresponding behaviours of P and Q satisfy S_P and S_Q, then we can deduce that $P \Lleftarrow\!\!\Vert\ Q$ satisfies S.

In [10] we investigated the relationship between the prioritized models and the timed failures model. We examined which timed failures could have resulted from a particular prioritized behaviour, and used this to produce an abstraction mapping between the two models. This not only helps us to understand the biased model, but also allows us to prove results about prioritized processes by proving corresponding results about their unprioritized abstractions.

In [11] we presented a proof system for the probabilistic language. We write $P \ \mathbf{sat}^{\geqslant p}\ S(\tau, \sqsubseteq, s)$ to mean that in every environment, the probability of P performing a behaviour that satisfies S is at least p. The proof rule for left-biased parallel composition is

$$\frac{\begin{array}{l} P \ \mathbf{sat}^{\geqslant p}\ S_P(\tau, \sqsubseteq, s) \\ Q \ \mathbf{sat}^{\geqslant q}\ S_Q(\tau, \sqsubseteq, s) \\ S_P(\tau, \sqsubseteq_P, s) \wedge S_Q(\tau, \sqsubseteq_Q, s) \Rightarrow S(\tau, \sqsubseteq_P \Lleftarrow\!\!\Vert \sqsubseteq_Q, s) \end{array}}{P \Lleftarrow\!\!\Vert\ Q \ \mathbf{sat}^{\geqslant pq}\ S(\tau, \sqsubseteq, s)}$$

Unfortunately, proving results about probabilistic processes is not as easy as in the unprobabilistic case. It turns out that we have to consider conditional specifications: we write $P \ \mathbf{sat}^{\geqslant p}\ S(\tau, \sqsubseteq, s) \mid S'(\tau, \sqsubseteq, s)$ if in all environments the probability that P performs a behaviour that satisfies S *given that* it satisfies S' is at least p. This and other factors make proofs involving probabilities rather complicated.

9.1 Acknowledgements

I would like to thank my supervisor, Bill Roscoe, for his advice and encouragement. Karen Seidel made many useful comments on an earlier version of this paper. I have also benefited from discussions with Tony Hoare, Steve Schneider and Jim Davies. This work was supported by a grant from the Science and Engineering Research Council of the United Kingdom and a scholarship from St. Hugh's College, Oxford.

References

[1] J. Camilleri. Introducing a priority operator to CCS. Technical report, Cambridge, 1989.

[2] R. Cleaveland and M. Hennessy. Priorities in process algebras. In *Proc 3rd Symposium on Logic in Computer Science, Edinburgh*, 1988.

[3] J. Davies. Specification and proof in real-time systems. Technical Monograph PRG-93, Oxford University Computing Labs, 1991.

[4] J. Davies and S. Schneider. A Brief History of Timed CSP. In this volume.

[5] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[6] G. Lowe. A probabilistic model of Timed CSP. D. Phil qualifying thesis, Oxford, 1991.

[7] G. Lowe. Prioritized and probabilistic models of Timed CSP. Technical Report PRG-TR-24-91, Oxford University Computing Labs, 1991.

[8] G. Lowe. Some extensions to the probabilistic, biased model of Timed CSP. Technical Report PRG-TR-9-92, Oxford University Computing Labs, 1992.

[9] G. Lowe. Specification and proof of prioritized, Timed CSP processes. Technical Report PRG-TR-17-92, Oxford University Computing Labs, 1992.

[10] G. Lowe. Relating the Prioritized Model of Timed CSP to the Timed Failures Model. Technical Report PRG-TR-18-92, Oxford University Computing Labs, 1992.

[11] G. Lowe. Specification and proof in probabilistic, prioritized, Timed CSP. Technical Report PRG-TR-23-92, Oxford University Computing Labs, 1992.

[12] G. Lowe. Representing Nondeterministic and Probabilistic Behaviour in Reactive Processes. Forthcoming, 1994.

[13] G. M. Reed. *A Uniform Mathematical Theory for Real-Time Distributed Computing.* D. Phil thesis, Oxford University Computing Labs, 1988.

[14] G. M. Reed and A. W. Roscoe. A timed model for CSP. In *ICALP86, LNCS 226*, pages 314–323. Springer Verlag, 1986.

[15] G. M. Reed and A. W. Roscoe. Metric spaces as models for real-time concurrency. In *MFPLS87*. Springer Verlag, 1987.

A Definitions of the operators

A.1 Subsidiary functions

$$
\begin{aligned}
\mathrm{fillout}\, f\ (\tau, \sqsubseteq, s) &\mathrel{\widehat{=}} \begin{cases} f(\tau, \sqsubseteq, s) & \text{if } (\tau, \sqsubseteq, s) \in \mathrm{dom}\, f \\ 0 & \text{if } (\tau, \sqsubseteq, s) \notin \mathrm{dom}\, f \end{cases} \\
\Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} w &\mathrel{\widehat{=}} \sqcup_{\sqsubseteq_P} \{ w'_P \in \mathrm{items}\, \sqsubseteq_P \mid w'_P \subseteq w \wedge w - w'_P \in \mathrm{items}\, \sqsubseteq_Q \} \\
&\qquad \text{if } \exists\, w_P \in \mathrm{items}\, \sqsubseteq_P, w_Q \in \mathrm{items}\, \sqsubseteq_Q \bullet w = w_P \uplus w_Q \\
\Uparrow^{g}_{\sqsubseteq} w &\mathrel{\widehat{=}} \sqcup_{\sqsubseteq} \{ w' \in \mathrm{items}\, \sqsubseteq \mid g w' = w \}, \qquad \text{if } \exists\, w' \in \mathrm{items}\, \sqsubseteq \bullet g w' = w
\end{aligned}
$$

A.2 Operations on offer relations

$$
\begin{aligned}
v(\sqsubseteq_P \mathbin{\square} \sqsubseteq_Q) w \;\Leftrightarrow\;& v \sqsubseteq_P w \\
& \vee\ w \in \mathrm{items}\, \sqsubseteq_P \wedge \Sigma w \neq \{\!|\,|\!\} \wedge v \in \mathrm{items}\, \sqsubseteq_Q \setminus \mathrm{items}\, \sqsubseteq_P \\
& \vee\ v, w \notin \mathrm{items}\, \sqsubseteq_P \wedge v \sqsubseteq_Q w
\end{aligned}
$$

$$
\begin{aligned}
v(\sqsubseteq_P {}^{X}\!\!\between^{Y} \sqsubseteq_Q)w \;\Leftrightarrow\;& (v \upharpoonright X \sqsubset_P w \upharpoonright X \vee v \upharpoonright X = w \upharpoonright X \wedge v \upharpoonright Y \sqsubseteq_Q w \upharpoonright Y) \\
& \wedge v \upharpoonright X, w \upharpoonright X \in \text{items} \sqsubseteq_P \wedge v \upharpoonright Y, w \upharpoonright Y \in \text{items} \sqsubseteq_Q \\
& \wedge \Sigma v, \Sigma w \subseteq X \cup Y \\
v(\sqsubseteq_P \between \sqsubseteq_Q)w \;\Leftrightarrow\;& \exists\, v'_P \in \text{items} \sqsubseteq_P, v'_Q \in \text{items} \sqsubseteq_Q \bullet v = v'_P \uplus v'_Q \\
& \wedge \exists\, w'_P \in \text{items} \sqsubseteq_P, w'_Q \in \text{items} \sqsubseteq_Q \bullet w = w'_P \uplus w'_Q \\
& \wedge v_P \sqsubset_P w_P \vee v_P = w_P \wedge v_Q \sqsubseteq_Q w_Q \\
& \quad \text{where } \; v_P = \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} v; \quad v_Q = v - v_P; \\
& \quad \phantom{\text{where }} \; w_P = \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} w; \;\; w_Q = w - w_P \\
v(\sqsubseteq \setminus X)w \;\Leftrightarrow\;& \exists\, v', w' \in \text{items} \sqsubseteq \bullet v' \setminus X = v \wedge w' \setminus X = w \wedge \Uparrow_{\sqsubseteq}^{\setminus X} v \sqsubseteq \Uparrow_{\sqsubseteq}^{\setminus X} w \\
v(g \odot \sqsubseteq)w \;\Leftrightarrow\;& \exists\, v' \in \text{items} \sqsubseteq \bullet gv' = v \wedge \exists\, w' \in \text{items} \sqsubseteq \bullet gw' = w \wedge \Uparrow_{\sqsubseteq}^{g} v \sqsubseteq \Uparrow_{\sqsubseteq}^{g} w
\end{aligned}
$$

A.3 Definitions of the operators

Let $A_P \mathrel{\widehat{=}} \mathcal{A}_{PBT}[\![P]\!]$, $A_Q \mathrel{\widehat{=}} \mathcal{A}_{PBT}[\![Q]\!]$, $f_P \mathrel{\widehat{=}} \mathcal{P}_{PBT}[\![P]\!]$, $f_Q \mathrel{\widehat{=}} \mathcal{P}_{PBT}[\![Q]\!]$.

$$\mathcal{A}_{PBT}[\![STOP]\!] \mathrel{\widehat{=}} \{(\tau, [0,\tau] \otimes \langle \{\!|\,|\!\}\rangle, \{\!|\,|\!\})\}$$

$$\mathcal{P}_{PBT}[\![STOP]\!] \mathrel{\widehat{=}} \text{fillout}\{(\tau, [0,\tau] \otimes \langle \{\!|\,|\!\}\rangle, \{\!|\,|\!\}) \mapsto 1\}$$

$$
\begin{aligned}
&\mathcal{A}_{PBT}[\![WAIT\ t]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, [0,\tau] \otimes \langle \{\!|\,|\!\}\rangle, \{\!|\,|\!\}) \mid \tau < t\} \\
&\quad \cup \{(\tau, [0,t) \otimes \langle \{\!|\,|\!\}\rangle ^\frown [t,\tau] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\,|\!\}\rangle, \prec\succ) \mid \tau \geqslant t\} \\
&\quad \cup \{(\tau, [0,t) \otimes \langle \{\!|\,|\!\}\rangle ^\frown [t,t'] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\,|\!\}\rangle ^\frown (t',\tau] \otimes \langle \{\!|\,|\!\}\rangle, \langle (t',\checkmark)\rangle) \mid t \leqslant t' \leqslant \tau\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{P}_{PBT}[\![WAIT\ t]\!] \mathrel{\widehat{=}} \\
&\quad \text{fillout}\,(\{(\tau, [0,\tau] \otimes \langle \{\!|\,|\!\}\rangle, \prec\succ) \mapsto 1 \mid \tau < t\} \\
&\qquad\qquad \cup \{(\tau, [0,t) \otimes \langle \{\!|\,|\!\}\rangle ^\frown [t,\tau] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\,|\!\}\rangle, \prec\succ) \mapsto 1 \mid \tau \geqslant t\} \\
&\qquad\qquad \cup \{(\tau, [0,t) \otimes \langle \{\!|\,|\!\}\rangle ^\frown [t,t'] \otimes \langle \{\!|\checkmark|\!\}, \{\!|\,|\!\}\rangle ^\frown (t',\tau] \otimes \langle \{\!|\,|\!\}\rangle, \langle (t',\checkmark)\rangle) \mapsto 1 \mid t \leqslant t' \leqslant \tau\})
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{A}_{PBT}[\![a \xrightarrow{0} P]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, [0,\tau] \otimes \langle \{\!|a|\!\}, \{\!|\,|\!\}\rangle, \prec\succ)\} \\
&\quad \cup \{(\tau, [0,t] \otimes \langle \{\!|a|\!\}, \{\!|\,|\!\}\rangle ^\frown \sqsubseteq_P + t, (t,a) ^\frown s_P + t) \mid (\tau - t, 0 \otimes \langle \{\!|\,|\!\}\rangle ^\frown \sqsubseteq_P, s_P) \in A_P \wedge \tau \geqslant t\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{P}_{PBT}[\![a \xrightarrow{0} P]\!] \mathrel{\widehat{=}} \\
&\quad \text{fillout}\,(\{(\tau, [0,\tau] \otimes \langle \{\!|a|\!\}, \{\!|\,|\!\}\rangle, \prec\succ) \mapsto 1\} \\
&\qquad\qquad \cup \{(\tau, [0,t] \otimes \langle \{\!|a|\!\}, \{\!|\,|\!\}\rangle ^\frown \sqsubseteq_P + t, (t,a) ^\frown s + t) \mapsto f_P(\tau - t, 0 \otimes \langle \{\!|\,|\!\}\rangle, \sqsubseteq_P, s) \mid \tau \geqslant t\})
\end{aligned}
$$

$$\mathcal{A}_{PBT}[\![P \ {}_p\sqcap_q\ Q]\!] \mathrel{\widehat{=}} \mathcal{A}_{PBT}[\![P]\!] \cup \mathcal{A}_{PBT}[\![Q]\!]$$

$$\mathcal{P}_{PBT}[\![P \ {}_p\sqcap_q\ Q]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} p.f_P(\tau, \sqsubseteq, s) + q.f_Q(\tau, \sqsubseteq, s)$$

$$
\begin{aligned}
&\mathcal{A}_{PBT}[\![P \mathbin{\square} Q]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q, \prec\succ) \mid (\tau, \sqsubseteq_P, \prec\succ) \in A_P \wedge (\tau, \sqsubseteq_Q, \prec\succ) \in A_Q\} \\
&\quad \cup \{(\tau, \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q, s) \mid \\
&\qquad\quad s \neq \prec\succ \wedge \mathit{begin}\ s = t \wedge (t, \sqsubseteq_P, \prec\succ) \in A_P \wedge (\tau, \sqsubseteq_Q, s) \in A_Q \wedge s \uparrow t \not\sqsupseteq_P (t, \{\!|\,|\!\})\} \\
&\quad \cup \{(\tau, \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q, s) \mid \\
&\qquad\quad s \neq \prec\succ \wedge \mathit{begin}\, s = t \wedge (\tau, \sqsubseteq_P, s) \in A_P \wedge (t, \sqsubseteq_Q, \prec\succ) \in A_Q \\
&\qquad\quad \wedge (s \uparrow t \sqsupset_P (t, \{\!|\,|\!\}) \vee s \uparrow t \notin \text{items} \sqsubseteq_Q)\}
\end{aligned}
$$

$$\mathcal{P}_{PBT}[\![P \mathbin{\square} Q]\!](\tau, \sqsubseteq, \prec\succ) \mathrel{\widehat{=}} \sum \{\!\!| f_P(\tau, \sqsubseteq_P, \prec\succ).f_Q(\tau, \sqsubseteq_Q, \prec\succ) \mid \sqsubseteq = \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q |\!\!\}$$

$$\begin{aligned}
&\mathcal{P}_{PBT}[\![P \mathbin{\square} Q]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \\
&\quad \sum \{\!\!| f_P(t, \sqsubseteq_P, \prec\succ).f_Q(\tau, \sqsubseteq_Q, s) \mid \sqsubseteq = \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q \wedge s \uparrow t \not\sqsupset_P (t, \{\!|\!\}) |\!\!\} \\
&\quad + \sum \{\!\!| f_P(\tau, \sqsubseteq_P, s).f_Q(t, \sqsubseteq_Q, \prec\succ) \mid \sqsubseteq = \sqsubseteq_P \mathbin{\square} \sqsubseteq_Q \wedge (s \uparrow t \sqsupset_P (t, \{\!|\!\}) \vee s \uparrow t \notin \mathrm{items} \sqsubseteq_Q) |\!\!\} \\
&\qquad \text{if } s \neq \prec\succ \wedge \mathit{begin}\ s = t
\end{aligned}$$

$$\begin{aligned}
&\mathcal{A}_{PBT}[\![P \mathbin{{}^{X}\Vert^{Y}} Q]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, \sqsubseteq_P \mathbin{{}^{X}\Vert^{Y}} \sqsubseteq_Q, s) \mid (\tau, \sqsubseteq_P, s \upharpoonright X) \in A_P \wedge (\tau, \sqsubseteq_Q, s \upharpoonright Y) \in A_Q \wedge \Sigma s \subseteq X \cup Y\}
\end{aligned}$$

$$\begin{aligned}
&\mathcal{P}_{PBT}[\![P \mathbin{{}^{X}\Vert^{Y}} Q]\!] \mathrel{\widehat{=}} \\
&\quad \mathrm{fillout}\,\{(\tau, \sqsubseteq, s) \mapsto \sum \{\!\!| f_P(\tau, \sqsubseteq_P, s \upharpoonright X).f_Q(\tau, \sqsubseteq_Q, s \upharpoonright Y) \mid \sqsubseteq = \sqsubseteq_P \mathbin{{}^{X}\Vert^{Y}} \sqsubseteq_Q |\!\!\} \mid \Sigma s \subseteq X \cup Y\}
\end{aligned}$$

$$\mathcal{A}_{PBT}[\![P \mathbin{|\!|\!|} Q]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq_P \mathbin{|\!|\!|} \sqsubseteq_Q, s) \mid (\tau, \sqsubseteq_P, \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} s) \in A_P \wedge (\tau, \sqsubseteq_Q, s - \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} s) \in A_Q\}$$

$$\mathcal{P}_{PBT}[\![P \mathbin{|\!|\!|} Q]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \sum \{\!\!| f_P(\tau, \sqsubseteq_P, \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} s).f_Q(\tau, \sqsubseteq_Q, s - \Uparrow_{\sqsubseteq_P, \sqsubseteq_Q} s) \mid \sqsubseteq_P \mathbin{|\!|\!|} \sqsubseteq_Q = \sqsubseteq |\!\!\}$$

$$\mathcal{A}_{PBT}[\![P \setminus X]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \forall t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq \wedge \exists \sqsubseteq' \bullet \sqsubseteq' \setminus X = \sqsubseteq \wedge (\tau, \sqsubseteq', \Uparrow^{\setminus X}_{\sqsubseteq'} s) \in A_P\}$$

$$\mathcal{P}_{PBT}[\![P \setminus X]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \sum \{\!\!| f_P(\tau, \sqsubseteq', \Uparrow^{\setminus X}_{\sqsubseteq'} s) \mid \sqsubseteq' \setminus X = \sqsubseteq |\!\!\}$$

$$\mathcal{A}_{PBT}[\![gP]\!] \mathrel{\widehat{=}} \{(\tau, \sqsubseteq, s) \mid \forall t \bullet s \uparrow t \in \mathrm{items} \sqsubseteq \wedge \exists \sqsubseteq' \bullet g \odot \sqsubseteq' = \sqsubseteq \wedge (\tau, \sqsubseteq', \Uparrow^{g}_{\sqsubseteq'} s) \in A_P\}$$

$$\mathcal{P}_{PBT}[\![gP]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \sum \{\!\!| f_P(\tau, \sqsubseteq', \Uparrow^{g}_{\sqsubseteq'} s) \mid g \odot \sqsubseteq' = \sqsubseteq |\!\!\}$$

$$\begin{aligned}
&\mathcal{A}_{PBT}[\![P \fatsemi Q]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, \sqsubseteq_P, s_P) \mid \forall t \bullet s_P \uparrow t \in \mathrm{items} \sqsubseteq_P \\
&\qquad\qquad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (\tau, \sqsubseteq'_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \checkmark \notin \Sigma(\Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P)\} \\
&\quad \cup \{(\tau, \sqsubseteq_P {}^\frown (t, \tau] \otimes \langle \{\!|\!\} \rangle, s_P) \mid \\
&\qquad t \leqslant \tau < t + \delta \wedge \forall t' \bullet s_P \uparrow t' \in \mathrm{items} \sqsubseteq_P \\
&\qquad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (t, \sqsubseteq'_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \mathit{begin}((\Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P) \upharpoonright \checkmark) = t\} \\
&\quad \cup \{(\tau, \sqsubseteq_P {}^\frown (t, t + \delta) \otimes \langle \{\!|\!\} \rangle {}^\frown \sqsubseteq_Q + t + \delta, s_P {}^\frown s_Q + t + \delta) \mid \\
&\qquad t \leqslant \tau - \delta \wedge \forall t' \bullet s_P \uparrow t' \in \mathrm{items} \sqsubseteq_P \\
&\qquad \wedge \exists \sqsubseteq'_P \bullet \sqsubseteq'_P \setminus \checkmark = \sqsubseteq_P \wedge (t, \sqsubseteq'_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P) \in A_P \wedge \mathit{begin}((\Uparrow^{\setminus \checkmark}_{\sqsubseteq'_P} s_P) \upharpoonright \checkmark) = t \\
&\qquad \wedge (\tau - (t + \delta), \sqsubseteq_Q, s_Q) \in A_Q\}
\end{aligned}$$

$$\begin{aligned}
&\mathcal{P}_{PBT}[\![P \fatsemi Q]\!](\tau, \sqsubseteq, s) \mathrel{\widehat{=}} \\
&\quad \sum \{\!\!| f_P(\tau, \sqsubseteq_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} s) \mid \sqsubseteq_P \setminus \checkmark = \sqsubseteq \wedge \checkmark \notin \Sigma(\Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} s) |\!\!\} \\
&\quad + \sum \{\!\!| f_P(t, \sqsubseteq_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} s) \mid \sqsubseteq = \sqsubseteq_P \setminus \checkmark {}^\frown (t, \tau |\!\!\} \otimes \langle \{\!|\!\} \rangle \wedge \mathit{begin}((\Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} s) \upharpoonright \checkmark) = t \wedge t \leqslant \tau < t + \delta] \\
&\quad + \sum \left\{\!\!\left| \begin{array}{l} f_P(t, \sqsubseteq_P, \Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} (s \trianglelefteq t)).f_Q(\tau - (t + \delta), \sqsubseteq_Q, s - t - \delta) \mid \\ \qquad \sqsubseteq = \sqsubseteq_P \setminus \checkmark {}^\frown (t, t + \delta) \otimes \langle \{\!|\!\} \rangle {}^\frown \sqsubseteq_Q + t + \delta \wedge \mathit{begin}((\Uparrow^{\setminus \checkmark}_{\sqsubseteq_P} s) \upharpoonright \checkmark) = t \wedge t \leqslant \tau - \delta \end{array} \right|\!\!\right\}
\end{aligned}$$

$$\begin{aligned}
&\mathcal{A}_{PBT}[\![\mathit{WAIT}\ t\ ; P]\!] \mathrel{\widehat{=}} \\
&\quad \{(\tau, [0, \tau] \otimes \langle \{\!|\!\} \rangle, \{\!|\!\}) \mid t > \tau\} \cup \{(\tau, \sqsubseteq + t, s + t) \mid t \leqslant \tau \wedge (\tau - t, \sqsubseteq, s) \in A_P\}
\end{aligned}$$

$$\begin{aligned}
&\mathcal{P}_{PBT}[\![\mathit{WAIT}\ t\ ; P]\!] \mathrel{\widehat{=}} \\
&\quad \mathrm{fillout}\,(\{(\tau, [0, \tau] \otimes \langle \{\!|\!\} \rangle, \{\!|\!\}) \mapsto 1 \mid t > \tau\} \\
&\qquad\quad \cup \{(\tau, \sqsubseteq, s) \mapsto f_P(\tau - t, \sqsubseteq - t, s - t) \mid t \leqslant \tau \wedge s \triangleleft t = \prec\succ \wedge \sqsubseteq \triangleleft t = [0, t) \otimes \langle \{\!|\!\} \rangle\})
\end{aligned}$$

$$\begin{aligned}
\mathcal{F}_{PBT}[\![\mu X \bullet F(X)]\!] \mathrel{\widehat{=}}\ & \text{the unique fixed point of the contraction mapping } C \circ M_\delta \\
& \text{where } C \text{ is the mapping on } PTM_B \text{ represented by } F
\end{aligned}$$

A.4 Derived operators

$$
\begin{array}{rcl}
SKIP & \mathrel{\widehat{=}} & WAIT\ 0 \\
a \xrightarrow{t} P & \mathrel{\widehat{=}} & a \xrightarrow{0} WAIT\ t\ ;P \\
P \mathbin{\square\!\!|} Q & \mathrel{\widehat{=}} & Q \mathbin{\square} P \\
P\ {}^{X}\!\!\mathrel{\|\!\!\!\rightarrow}{}^{Y}\ Q & \mathrel{\widehat{=}} & Q\ {}^{Y}\!\!\mathrel{\leftarrow\!\!\!\|}{}^{X}\ P \\
P \mathrel{\leftarrow\!\!\!\|} Q & \mathrel{\widehat{=}} & P\ {}^{\Sigma}\!\!\mathrel{\leftarrow\!\!\!\|}{}^{\Sigma}\ Q \\
P \mathrel{\|\!\!\!\rightarrow} Q & \mathrel{\widehat{=}} & Q\ {}^{\Sigma}\!\!\mathrel{\leftarrow\!\!\!\|}{}^{\Sigma}\ P \\
P \mathrel{-\!\!\!|\!|\!|\!\!\!\rightarrow} Q & \mathrel{\widehat{=}} & Q \mathrel{\leftarrow\!\!\!|\!|\!|\!\!\!-} P
\end{array}
$$