Fault-Tolerant Bisimulation and Process Transformations *

Tomasz Janowski **

Department of Computer Science
University of Warwick, Coventry CV4 7AL, UK

Abstract. We provide three methods of verifying concurrent systems which are tolerant of faults in their operating environment - algebraic, logical and transformational. The first is an extension of the bisimulation equivalence, the second is rooted in the Hennessy-Milner logic, and the third involves transformations of CCS processes. Based on the common semantic model of labelled transition systems, which is also used to model faults, all three methods are proved equivalent for certain classes of faults.

1 Introduction

Many models of concurrent systems have been proposed in the literature, based on either actions or states. Examples include sequences [MP91], trees [Mil89], machines [LT87], partial orders [Pra86] and event structures [Win89]. They offer different ways of representing executions of systems (linear or branching), their concurrent activity (interleaving or non-interleaving) and interaction (shared memory or message-passing). A concept which unifies various models is a *labelled transition system* [Kel76], a triple $(\mathcal{P}, \mathcal{A}, \rightarrow)$ where $\mathcal{P}$ is a set of processes, $\mathcal{A}$ a set of actions and $\rightarrow \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{P}$ a labelled transition relation. Labelled transition relations are often defined by induction on the structure of processes, providing the *structured operational semantics* [Plo81] of process description languages. An example of such a language is CCS [Mil89].

As models of processes, labelled transition systems describe their behaviour in detail, including particulars of their internal computation. However, in order to specify a process and then to prove its correctness, it is useful to decide which properties of the model are relevant and which can be ignored. Following [Mil89], it is most common to ignore these properties which cannot be observed in the finite interval of time. Two ways to do so are as follows:

- we can identify a process with its equivalence class, according to the (weak) *bisimulation equivalence* $\approx$ [Par81];
- we can identify a process with its properties, specified by the (weak) formulas of the Hennessy-Milner logic [HM85] and verified by satisfaction relation $\models$.

* To be presented at the Third International Symposium "Formal Techniques in Real-Time and Fault-Tolerant Systems", Lübeck, Germany, September 1994.

** Supported by the University of Warwick, under its Scholarship Scheme for East Europe, and by an Overseas Students Award from CVCP.

A vital test of the usefulness of any formal theory is that statements of this theory must be confirmed in practice (by experiment). Given such statements as $P \approx Q$ or $Q \models M$, it is expected that the low-level process Q, when placed in the real environment, behaves respectively as specified by the high-level process P or the formula M. In practice however, such Q depends on various hardware components which often malfunction because of the physical faults. Such faults affect the semantics of Q so that it may no longer behave as specified. Moreover, physical faults do not exist before Q is put into practice and so cannot be removed beforehand, they must be *tolerated.*

Clearly, it is not possible to tolerate arbitrary faults. We have to decide which faults are *anticipated* (and thus should be tolerated) and which are not (such faults are *catastrophical*). To represent the effect of the anticipated faults on the semantics of processes, we will use the set $\dashrightarrow$ of the *faulty transitions*. To verify fault-tolerance is then to prove that the low-level process behaves 'correctly' in the presence of the transitions $\dashrightarrow$. As such, fault-tolerance depends on the chosen notion of correctness. In this paper we provide three methods to verify fault-tolerance for bisimulation equivalence and the Hennessy-Milner logic:

1. A fault-tolerant bisimilarity $\boxdot$ where $P \boxdot Q$ if observing P in the fault-free environment (performing transitions $\longrightarrow$) and Q in the environment which contains anticipated faults (performing transitions $\longrightarrow$ and $\dashrightarrow$), we cannot tell them apart in the finite interval of time.
2. A relation $\Vdash$ to verify satisfaction of formulas of the Hennessy-Milner logic in the presence of the anticipated faults (when processes $\mathcal{P}$ undergo both normal and faulty transitions $\longrightarrow \cup \dashrightarrow$).
3. A language $\mathcal{D}$ for specifying faults and a process transformation $\mathcal{T}(Q, \Psi)$ where given the CCS process Q, the effect of transitions $\overset{}{\underset{\Psi}{\dashrightarrow}}$ (specified by $\Psi \in \mathcal{D}$) on Q are represented syntactically. Then, verifying that Q is fault-tolerant involves proving either:
   (a) $P \approx \mathcal{T}(Q, \Psi)$ for the high-level process P, or
   (b) $\mathcal{T}(Q, \Psi) \models M$ for the formula M.

We show that, for wide classes of faults, all these methods are equivalent:

$$\begin{array}{ccc} P \boxdot Q & \longleftrightarrow & \forall_{M \in \mathcal{M}}\, P \models M \text{ iff } Q \Vdash M \\ \updownarrow & & \updownarrow \\ P \approx \mathcal{T}(Q, \Psi) & \longleftrightarrow & \forall_{M \in \mathcal{M}}\, P \models M \text{ iff } \mathcal{T}(Q, \Psi) \models M \end{array} \tag{1}$$

When the 'full' fault-tolerance is either impossible or too expensive to ensure, we may be still satisfied with its conditional version, given certain assumption about the quantity of faults. To this end we will use $n \in \mathbb{N} \cup \{\infty\}$ as the maximal number of times transitions $\dashrightarrow$ can occur successively (if $n = \infty$ then $\dashrightarrow$ can occur at any time; if $n = 0$ then not at all). As before, we provide and prove the equivalence of the three methods for verifying n-conditional fault-tolerance: relations $\boxdot_n$ and $\Vdash_n$, and transformation $\widetilde{\mathcal{T}}(\ .\ , \Psi, n)$.

The rest of this paper is as follows. In Section 2 we describe the semantic model. 'Intolerant' bisimulation equivalence, its fault-tolerant and conditional fault-tolerant versions are defined in Section 3. Their logical characterisation, in terms of the Hennessy-Milner logic, is given in Section 4. Both languages, of processes and faults, are defined in Sections 5 and 6, followed by transformations $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ which are shown to provide the third, equivalent method of verifying fault-tolerance and conditional fault-tolerance in Section 7. Finally, in Section 8, we draw some conclusions and comment on the directions for future work.

2 Semantic Model

Consider the labelled transition relation $\longrightarrow$. If $(P, \alpha, P') \in \longrightarrow$ then we write $P \xrightarrow{\alpha} P$ and say that P performs α and evolves into P' (we also use $P \xrightarrow{s} P'$ for the action sequence $s \in \mathcal{A}^*$). One kind of transition we wish to largely ignore is $\xrightarrow{\tau}$ where action τ is unobservable and represents the outcome of a joint activity (interaction) between two processes. Interaction takes place on the pair of complementary actions $a, \overline{a} \in \mathcal{L}$ where $\mathcal{L} =_{def} \mathcal{A} - \{\tau\}$ is the set of observable actions and $\overline{\alpha}$ is a function over $\mathcal{A}$ which is bijective and such that $\overline{\overline{a}} = a$, $\overline{a} \neq \tau$ and $\overline{\tau} = \tau$. We use α to range over $\mathcal{A}$ and β to range over $\mathcal{L}_\varepsilon =_{def} \mathcal{L} \cup \{\varepsilon\}$ where ε denotes the empty sequence. We also let $\widehat{\cdot}$ be the function over $\mathcal{A}^*$ such that $\widehat{\varepsilon} = \varepsilon$, $\widehat{\tau : s} = \widehat{s}$ and $\widehat{a : s} = a : \widehat{s}$ (: denotes concatenation).

When placed in the 'real' environment, a process may not behave according to $\longrightarrow$: it may either perform transitions $\dashrightarrow$ which do not belong to $\longrightarrow$, $\dashrightarrow \cap \longrightarrow = \emptyset$, or it may refuse to perform some of the transitions $\longrightarrow$. The first case is demonstrated by transition $P \overset{c}{\dashrightarrow} P''$ in Figure 1, the second (in part) by $Q \xrightarrow{b} Q'''$. To represent the second case in full, we should physically remove $Q \xrightarrow{b} Q'''$ from the diagram. This would complicate our model so suppose only that this transition *may be* refused. This is achieved by two more transitions $Q \overset{\tau}{\dashrightarrow} Q'' \overset{a}{\dashrightarrow} Q'$ which may preempt (due to the occurrence of τ) transition $Q \xrightarrow{b} Q'''$. Given the set $\dashrightarrow$ of faulty transitions as the effect of faults, we let $\longmapsto$ contain both kinds of transitions, $\longmapsto =_{def} \longrightarrow \cup \dashrightarrow$ (we also use $\overset{s}{\longmapsto}$ for the action sequence $s \in \mathcal{A}^*$).

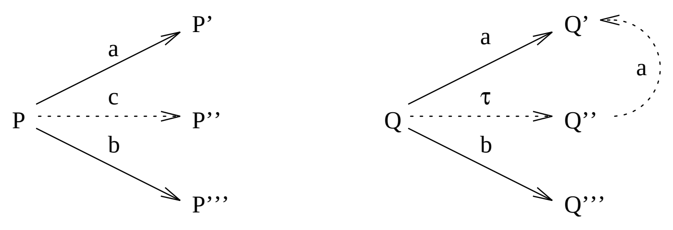

Fig. 1. Transition diagrams of P and Q with faulty transitions.

3 Bisimulation and Fault-Tolerance

There are many equivalences by which to abstract from the behavioural details of the transition relation $\longrightarrow$. They differ, among other things, in the adopted model of execution (linear- or branching-time) and concurrency (interleaving or non-interleaving). The best known of them and deemed to be the strongest among interleaving and branching-time equivalences is that of bisimulation equivalence [Par81], $\approx$. Two bisimilar processes, their semantics defined by transition relation $\longrightarrow$, cannot be distinguished by observing them in the finite interval of time. This property may no longer hold in the presence of faults which result in the additional transitions $\dashrightarrow$ (of the low-level process). As such, $\approx$ allows to verify correctness in the absence of faults only, it is *fault-intolerant*. How to verify *fault-tolerance*, i.e. correctness in the presence of transitions $\dashrightarrow$, and *conditional fault-tolerance*, where transitions $\dashrightarrow$ occur under assumption n about their quantity, is the topic of the current section.

3.1 Fault-Intolerance

Bisimulation equivalence is defined as the maximal fixed point of the functional $\mathcal{F}$ on the set of binary relations B on $\mathcal{P}$, $(P, Q) \in \mathcal{F}(B)$ iff

$$\begin{array}{l} \text{whenever } P \stackrel{\alpha}{\longrightarrow} P' \text{ then } \exists_{Q',s}\ Q \stackrel{s}{\longrightarrow} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B \\ \text{whenever } Q \stackrel{\alpha}{\longrightarrow} Q' \text{ then } \exists_{P',s}\ P \stackrel{s}{\longrightarrow} P' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B \end{array} \qquad (2)$$

This maximal fixed point exists because $\mathcal{F}$ is monotonic: if $B_1 \subseteq B_2$ then $\mathcal{F}(B_1) \subseteq \mathcal{F}(B_2)$. Originally, it was reached 'from above', as the limit of the sequence $\mathcal{F}^n(\mathcal{P} \times \mathcal{P})$ for all $n \geq 0$. Unfortunately, unless infinite ordinals n are taken into account, this requires that transition relations $\stackrel{\alpha}{\longrightarrow}$ are weak-image-finite i.e. that for all $P \in \mathcal{P}$, the set $\{P' \mid P \stackrel{s}{\longrightarrow} P' \ \wedge\ \widehat{s} = \widehat{\alpha}\}$ is finite. No such assumption is needed to reach $\approx$ 'from below', as the union of all pre-fixed points B of $\mathcal{F}$, $\approx =_{def} \bigcup\{B \mid B \subseteq \mathcal{F}(B)\}$ [Par81]. An additional advantage is the useful technique for proving $P \approx Q$. It is enough to find a pre-fixed point B of $\mathcal{F}$ such that $(P, Q) \in B$. Such a B is called a *bisimulation*. We have:

$$\begin{array}{rl} P \approx Q \text{ iff} & \text{whenever } P \stackrel{\alpha}{\longrightarrow} P' \text{ then } \exists_{Q',s}\ Q \stackrel{s}{\longrightarrow} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge P' \approx Q' \\ & \text{whenever } Q \stackrel{\alpha}{\longrightarrow} Q' \text{ then } \exists_{P',s}\ P \stackrel{s}{\longrightarrow} P' \wedge \widehat{s} = \widehat{\alpha} \wedge P' \approx Q' \end{array} \qquad (3)$$

3.2 Fault-Tolerance

If observing two processes, the high-level in the fault-free environment (performing transitions $\longrightarrow$) and the lower-level in the environment which is affected by the anticipated faults (performing transitions $\longmapsto$), we cannot tell them apart in the finite interval of time, then we say that the lower-level process is fault-tolerant (with respect to the high-level one). To verify this property we provide two relations, *must-bisimilarity* $\boxed{!}$ and *may-bisimilarity* $\boxed{?}$.

The first is the direct extension of $\approx$ to take account of transitions $\dashrightarrow$. We have $P \sqsubseteq_{!} Q$ if P and Q are 'bisimilar', the first performing transitions $\longrightarrow$ and the second both $\longrightarrow$ and $\dashrightarrow$. We define $\sqsubseteq_{!}$ using a *must-bisimulation* B which is a binary relation such that if $(P,Q) \in B$ then any $\longrightarrow$ transition of P is matched by some transition sequence $\longmapsto$ of Q and any $\longmapsto$ transition of Q is matched by some transition sequence $\longrightarrow$ of P, such that the matched transitions have the same observable actions and B is preserved:

$$\begin{array}{l} \text{whenever } P \xrightarrow{\alpha} P' \text{ then } \exists_{Q',s}\ Q \xmapsto{s} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge (P',Q') \in B \\ \text{whenever } Q \xmapsto{\alpha} Q' \text{ then } \exists_{P',s}\ P \xrightarrow{s} P' \wedge \widehat{s} = \widehat{\alpha} \wedge (P',Q') \in B \end{array} \tag{4}$$

Then $P \sqsubseteq_{!} Q$ iff $(P,Q) \in B$ for some must-bisimulation B. Such Q satisfies the basic postulate: no external observer can distinguish between P which behaves according to transitions $\longrightarrow$ and Q which may additionally perform transitions $\dashrightarrow$. In one aspect however, such Q is unsatisfactory. Because $\sqsubseteq_{!}$ allows to match transitions of the high-level process by faulty transitions of the low-level one, such Q may not behave properly in the environment where not all transitions $\dashrightarrow$ are provided. For Q to behave as specified, transitions $\dashrightarrow$ *must* occur. In practice however, it is more useful is to assume the mere possibility of faults (that faults *may* occur), not their necessity (that they *must* occur).

This assumption is met by may-bisimilarity $P \sqsubseteq_{?} Q$ where only normal transitions of Q are allowed to match transitions of P. As before, may-bisimilarity $\sqsubseteq_{?}$ is defined as the largest *may-bisimulation* which is a binary relation B such that if $(P,Q) \in B$ then:

$$\begin{array}{l} \text{whenever } P \xrightarrow{\alpha} P' \text{ then } \exists_{Q',s}\ Q \xrightarrow{s} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge (P',Q') \in B \\ \text{whenever } Q \xmapsto{\alpha} Q' \text{ then } \exists_{P',s}\ P \xrightarrow{s} P' \wedge \widehat{s} = \widehat{\alpha} \wedge (P',Q') \in B \end{array} \tag{5}$$

Example 1. Consider the high-level process P in Figure 2 and four low-level, fault-affected processes Q, R, S and T. We have:

1. $P \approx Q$ because $\{(P,Q),(P',Q'),(P',Q'')\}$ is a bisimulation but $P \not\sqsubseteq_{!} Q$ and $P \not\sqsubseteq_{?} Q$ because there is no must- or may-bisimulation which contains (P,Q): $Q \xmapsto{\tau} Q'''$ and $P \xrightarrow{s} P$ ($\widehat{s} = \varepsilon$) only, however $P \xrightarrow{a} P'$ but $Q''' \not\longmapsto$.
2. $P \sqsubseteq_{!} R$ because $\{(P,R),(P,R'''),(P',R'),(P',R'')\}$ is a must-bisimulation but $P \not\approx R$ and $P \not\sqsubseteq_{?} R$ because there is no bisimulation or may-bisimulation which contains (P,R): $P \xrightarrow{a} P'$ and $R \xrightarrow{s} R'$ ($\widehat{s} = a$) only, however $P' \xrightarrow{b} P$ but R' has no normal transitions, $R' \not\longrightarrow$.
3. $P \sqsubseteq_{!} S$ because $\{(P,S),(P,S'''),(P',S'),(P',S'')\}$ is a must-bisimulation and $P \approx S$ because $\{(P,S),(P',S'),(P',S'')\}$ is a bisimulation. Also $P \not\sqsubseteq_{?} S$ because there is no may-bisimulation which contains (P,S): $S \xmapsto{\tau} S'''$ and $P \xrightarrow{s} P$ ($\widehat{s} = \varepsilon$) only, however $P \xrightarrow{a} P'$ but $S''' \overset{a}{\not\longrightarrow}$.
4. $P \sqsubseteq_{!} T$, $P \sqsubseteq_{?} T$ and $P \approx T$ because $\{(P,T),(P,T'''),(P',T'),(P',T'')\}$ is simultaneously a must-bisimulation, may-bisimulation and bisimulation. $\square$

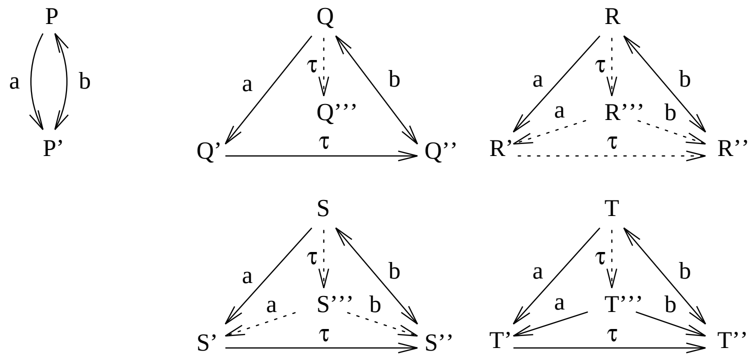

Fig. 2. Transition diagrams of P, Q, R, S and T with faulty transitions.

The example shows that $\approx$ and $\sqsubseteq_!$ are not comparable: $\sqsubseteq_!$ does not imply $\approx$, nor does $\approx$ imply $\sqsubseteq_!$. However, it is easy to show that any may-bisimulation is simultaneously a must-bisimulation and a bisimulation. As a result, because all relations are defined as the union of the corresponding bisimulations, we have:

$$\sqsubseteq_? \quad \subseteq \quad \sqsubseteq_! \cap \approx \tag{6}$$

The example ($P \sqsubseteq_! S$ and $P \approx S$ but $P \not\sqsubseteq_? S$) also shows that this inclusion is proper i.e. that $P \sqsubseteq_! Q$ and $P \approx Q$ together are not enough to establish $P \sqsubseteq_? Q$. That is a pity since $P \sqsubseteq_? Q$ which is more desirable than $P \sqsubseteq_! Q$, is also more difficult to establish (the equivalence diagram (1) for $\sqsubseteq_!$ is only partly valid for may-bisimilarity $\sqsubseteq_?$). However, for B to be a may-bisimulation, it is not only necessary but also sufficient that B is a bisimulation and a must-bisimulation:

$$B \text{ is a may-bisimulation iff it is a must-bisimulation and a bisimulation.} \tag{7}$$

Thus in order to prove $P \sqsubseteq_? Q$, it is enough to show that $(P, Q) \in B$ for B which is a bisimulation and a must-bisimulation at the same time. This justifies our efforts to establish the properties of $\sqsubseteq_!$ in the first instance. What both relations have in common is that neither of them is reflexive or symmetric (they are not preorders). For processes in Figure 2 we have:

- $Q \not\sqsubseteq_! Q$ because there is no must-bisimulation which contains the pair (Q, Q): $Q \overset{\tau}{\longmapsto} Q'''$ and $Q \overset{s}{\longrightarrow} Q$ ($\widehat{s} = \varepsilon$) only, however $Q \overset{a}{\longrightarrow} Q'$ but $Q''' \not\longmapsto$. Consequently $Q \not\sqsubseteq_? Q$ because of (6).
- $Q \sqsubseteq_? P$ because $\{(Q, P), (Q', P'), (Q'', P')\}$ is a may-bisimulation but $P \not\sqsubseteq_! Q$ because $Q \overset{\tau}{\longmapsto} Q'''$ and $P \overset{s}{\longrightarrow} P$ ($\widehat{s} = \varepsilon$) only, however $P \overset{a}{\longrightarrow} P'$ but $Q''' \not\longmapsto$. Consequently we have $Q \sqsubseteq_? P$ and $P \not\sqsubseteq_? Q$, as well as $Q \sqsubseteq_! P$ and $P \not\sqsubseteq_! Q$.

The lack of these properties is not unexpected when verifying correctness in the presence of faults. Because one and the same process has two different semantics, as the high-level (fault-free) process and as the low-level (fault-affected) one, we cannot ensure that the underlying relation is reflexive or symmetric.

Transitivity is most desirable to support the stepwise development of processes and to support the reasoning in the presence of faults where it may be helpful to deal with only some of transitions $\dashrightarrow$ (not all) at a time. To this end let us partition $\dashrightarrow$ among $m > 0$ nonempty disjoint sets $\underset{j}{\dashrightarrow}$, $\dashrightarrow\ = \bigcup_{j=1}^{m} \underset{j}{\dashrightarrow}$. Given $j = 0, \ldots, m$ we define $\xmapsto[j]{}$ as the union of the normal transitions $\longrightarrow$ and the first j partitions of $\dashrightarrow$: $\xmapsto[j]{}\ =_{def}\ \longrightarrow\ \cup \bigcup_{i=1}^{j} \underset{i}{\dashrightarrow}$. This gives an ascending sequence $\longrightarrow\ =\ \xmapsto[0]{}\ \subseteq \cdots \subseteq\ \xmapsto[m]{}\ =\ \longmapsto$ of the transition relations.

Suppose now that for $0 \leq j \leq l \leq m$, $\boxed{?}_{l}^{j}$ and $\boxed{!}_{l}^{j}$ denote the corresponding bisimulation relations where transitions $\xmapsto[j]{}$ are regarded as normal and transitions $\bigcup_{i=j+1}^{l} \underset{i}{\dashrightarrow}$ as abnormal:

$$\begin{aligned} P \boxed{?}_{l}^{j} Q \text{ iff whenever } P \xmapsto[j]{\alpha} P' \text{ then } \exists_{Q',s}\ Q \xmapsto[j]{s} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge P' \boxed{?}_{l}^{j} Q' \\ \text{whenever } Q \xmapsto[l]{\alpha} Q' \text{ then } \exists_{P',s}\ P \xmapsto[j]{s} P' \wedge \widehat{s} = \widehat{\alpha} \wedge P' \boxed{?}_{l}^{j} Q' \end{aligned} \tag{8}$$

Relation $\boxed{!}_{l}^{j}$ is defined alike. Then, given $j \leq k \leq l$, we can easily prove the following transitive properties of $\boxed{!}$ and $\boxed{?}$ ($\circ$ is the relational composition):

$$\boxed{?}_{k}^{j} \circ \boxed{?}_{l}^{j} \subseteq \boxed{?}_{l}^{j} \qquad\qquad \boxed{!}_{k}^{j} \circ \boxed{!}_{l}^{k} \subseteq \boxed{!}_{l}^{j} \tag{9}$$

According to the first inclusion, to tolerate transitions $\bigcup_{i=j+1}^{l} \underset{i}{\dashrightarrow}$ (given $\boxed{?}$), at least once we must tolerate them altogether. According to the second, to tolerate transitions $\bigcup_{i=j+1}^{l} \underset{i}{\dashrightarrow}$ (with respect to $\boxed{!}$), it is enough to first tolerate transitions $\bigcup_{i=j+1}^{k} \underset{i}{\dashrightarrow}$ and then transitions $\bigcup_{i=k+1}^{l} \underset{i}{\dashrightarrow}$. Following the first inclusion, it is easy to see that $\boxed{?}$ is transitive. This is not the case for $\boxed{!}$ and in general the first inclusion does not hold for $\boxed{!}$ and the second for $\boxed{?}$, as shown by processes in Figure 3. We have: $P \boxed{!} Q \boxed{!} R$ but clearly $P \not\boxed{} R$. Also, $P \boxed{?}_{1}^{0} S$ and $S \boxed{?}_{2}^{1} T$ but $P \not\boxed{?}_{2}^{0} T$ because $T \overset{\tau}{\underset{2}{\dashrightarrow}} T'''$ and $P \xmapsto[0]{s} P$ ($\widehat{s} = \varepsilon$) only, however $P \xmapsto[0]{a} P'$ but $T''' \not\xmapsto[0]{}$.

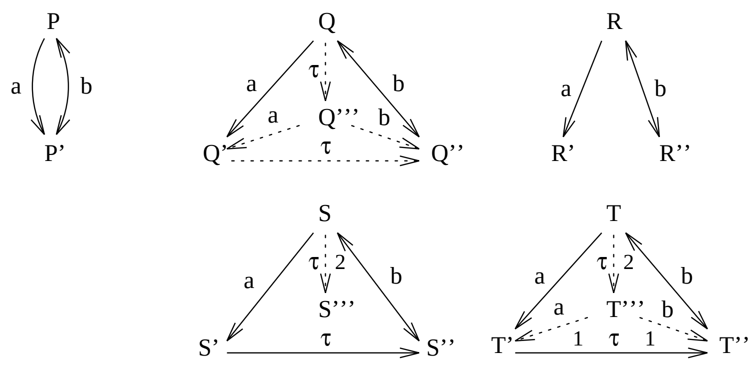

Fig. 3. Transition diagrams of P, Q, R, S and T with faulty transitions.

3.3 Conditional Fault-Tolerance

For any may- or must-bisimilar processes P and Q, $\dashrightarrow$ is the assumption about faults of the operating environment of Q, where Q is guaranteed to behave 'properly', as specified by P. We call $\dashrightarrow$ a *qualitative* assumption, in opposite to the *quantitative* assumptions $n \in \mathbb{N} \cup \{\infty\}$ which are introduced in this section and specify the maximal number of times transitions $\dashrightarrow$ can occur successively (if $n = 0$ then $\dashrightarrow$ are assumed not to occur at all; if $n = \infty$ then they can occur at any time). The reasons for introducing such assumptions are threefold:

- For certain sets $\dashrightarrow$, we cannot ensure fault-tolerance in full. In these circumstances, we must be satisfied with its degraded, conditional version, for certain assumptions about the quantity of $\dashrightarrow$.
- Even when the 'full' fault-tolerance is (in theory) possible, we may choose its conditional version because it is often easier to do so. This argument is true for applications which are not safety-critical.
- Conditional fault-tolerance may facilitate the stepwise procedure where Q is first designed for restricted assumptions about faults and then stepwise transformed for increasingly relaxed assumptions.

Recall that if $Q \overset{s}{\longmapsto} Q'$ then Q evolves into Q' performing the sequence s of transitions $\longrightarrow$ and $\dashrightarrow$. This may be no longer the case if transitions $\dashrightarrow$ can only occur under assumption n. It these circumstances we will use the family $\{\longmapsto^i_j\}^n_{i,j=0}$ of relations $\longmapsto^i_j \subseteq \mathcal{P} \times \mathcal{A}^* \times \mathcal{P}$. If $(Q, s, Q') \in \longmapsto^i_j$ then we write $Q \overset{s}{\longmapsto}{}^i_j Q'$ and say that Q evolves into Q' by the sequence $s \in \mathcal{A}^*$ of transitions $\longmapsto$, under assumption n, and given that i is the number of times transitions $\dashrightarrow$ have successively occurred before and j after $Q \overset{s}{\longmapsto}{}^i_j Q'$. Formally, relations $\longmapsto^i_j$ are defined by the following inductive rules:

$$\begin{array}{ll} Q \overset{\varepsilon}{\longmapsto}{}^i_i Q & \\ Q \overset{\alpha:s}{\longmapsto}{}^i_j Q'' \text{ whenever } \exists_{Q'} & Q \overset{\alpha}{\longrightarrow} Q' \overset{s}{\longmapsto}{}^0_j Q'' \quad \vee \\ & Q \overset{\alpha}{\dashrightarrow} Q' \overset{s}{\longmapsto}{}^{i+1}_j Q'' \wedge i \neq n \end{array} \tag{10}$$

The induction above is well-defined: the first rule provides the base, for the empty sequence ε, and the second rule decreases the length of the action sequence by one. Given $n = 0$, we always have $i = n$, so transitions $\dashrightarrow$ cannot occur at all. Given $n = \infty$, it is never the case that $i = n$, so $\dashrightarrow$ can occur at any time.

Consider conditional version of $\boxed{?}$, $\boxed{?}_n$. We have $P \boxed{?}_n Q$ if observing P in the fault-free environment (performing transitions $\longrightarrow$) and Q in any environment where it *may* also perform transitions $\dashrightarrow$ (provided no more than n times in a row), we cannot distinguish between them in the finite amount of time. In order to keep track of the number i of the successive transitions $\dashrightarrow$, $\boxed{?}_n$ is defined using $[0, n]$-indexed families $\{B_i\}^n_{i=0}$ of binary relations $B_i \subseteq \mathcal{P} \times \mathcal{P}$. Such a family $\{B_i\}^n_{i=0}$ is called a conditional must-bisimulation iff for all $i, j \in [0, n]$ and $\alpha \in \mathcal{A}$, if $(P, Q) \in B_i$ then:

$$\begin{array}{l} \text{whenever } P \overset{\alpha}{\longrightarrow} P' \text{ then } \exists_{Q',s,j}\ Q \overset{s}{\longmapsto}{}^i_j Q' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B_j \\ \text{whenever } Q \overset{\alpha}{\longmapsto}{}^i_j Q' \text{ then } \exists_{P',s}\ \ P \overset{s}{\longrightarrow} P' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B_j \end{array} \tag{11}$$

Similarly, $\{B_i\}_{i=0}^n$ is called a conditional may-bisimulation iff for all $i, j \in [0, n]$ and $\alpha \in \mathcal{A}$, if $(P, Q) \in B_i$ then:

$$\begin{array}{l} \text{whenever } P \xrightarrow{\alpha} P' \text{ then } \exists_{Q',s}\ Q \xrightarrow{s} Q' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B_i \\ \text{whenever } Q \overset{\alpha}{\longmapsto}{}^{i}_{j} Q' \text{ then } \exists_{P',s}\ P \xrightarrow{s} P' \wedge \widehat{s} = \widehat{\alpha} \wedge (P', Q') \in B_j \end{array} \tag{12}$$

Let $i \in [0, n]$. We define relations $\overset{!}{\sqsubseteq}{}^{i}_{n}$ and $\overset{?}{\sqsubseteq}{}^{i}_{n}$ as follows:

$$\begin{array}{ll} P \overset{!}{\sqsubseteq}{}^{i}_{n} Q \text{ iff } (P, Q) \in B_i & \text{for some } \{B_i\}_{i=0}^n \\ & \text{which is a conditional must-bisimulation} \\ P \overset{?}{\sqsubseteq}{}^{i}_{n} Q \text{ iff } (P, Q) \in B_i & \text{for some } \{B_i\}_{i=0}^n \\ & \text{which is a conditional may-bisimulation} \end{array}$$

Then we have $P \overset{!}{\sqsubseteq}_{n} Q$ iff $P \overset{!}{\sqsubseteq}{}^{0}_{n} Q$ and $P \overset{?}{\sqsubseteq}_{n} Q$ iff $P \overset{?}{\sqsubseteq}{}^{0}_{n} Q$.

Example 2. Consider processes in Figure 4. We have:

- $P \not\overset{?}{\sqsubseteq} Q$ and $P \not\overset{?}{\sqsubseteq}_2 Q$ because of transition $Q''' \overset{a}{\dashrightarrow} Q$ which enables two subsequent actions a. However $P \overset{?}{\sqsubseteq}_1 Q$ because then $Q''' \overset{a}{\dashrightarrow} Q$ (as the second one in a row) cannot be chosen.
- $P \not\overset{!}{\sqsubseteq} R$ and $P \not\overset{!}{\sqsubseteq}_3 R$ because of transition $R''' \overset{\tau}{\dashrightarrow} R''$ which, after performing a and b, leads to R'' where action a is not possible. However $P \overset{!}{\sqsubseteq}_2 R$ because then transition $R''' \overset{\tau}{\dashrightarrow} R''$ (as the third one in a row) cannot be chosen. Finally $P \not\overset{!}{\sqsubseteq}_1 R$ because then $R' \overset{b}{\dashrightarrow} R'''$ cannot be taken but is needed to match transitions of P. $\square$

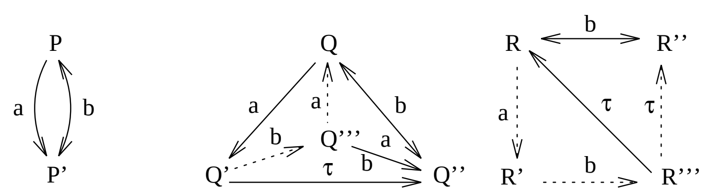

Fig. 4. Transition diagrams of P, Q and R with abnormal transitions.

Conditional may-bisimilarity $\overset{?}{\sqsubseteq}_n$ is monotone decreasing with respect to n. This is not the case for $\overset{!}{\sqsubseteq}_n$, as shown by $P \overset{!}{\sqsubseteq}_2 R$ and $P \not\overset{!}{\sqsubseteq}_1 R$ in Figure 4. For $n = \infty$, $\overset{!}{\sqsubseteq}_\infty$ and $\overset{?}{\sqsubseteq}_\infty$ coincide with their unconditional versions; for $n = 0$, they coincide with $\approx$. We have the following diagram of inclusions:

$$\begin{array}{ccccccccc} \overset{?}{\sqsubseteq} = \overset{?}{\sqsubseteq}_\infty & \subset \cdots \subset & \overset{?}{\sqsubseteq}_{n+1} & \subset & \overset{?}{\sqsubseteq}_n & \subset \cdots \subset & \overset{?}{\sqsubseteq}_0 = \approx \\ \cap & & \cap & & \cap & & \| \\ \overset{!}{\sqsubseteq} = \overset{!}{\sqsubseteq}_\infty & & \overset{!}{\sqsubseteq}_{n+1} & & \overset{!}{\sqsubseteq}_n & & \overset{!}{\sqsubseteq}_0 = \approx \end{array} \tag{13}$$

4 Logic of Processes

The Hennessy-Milner logic [HM85] is a simple modal logic for specifying properties of processes. It provides a language $\mathcal{M}$ of formulas M which extends propositional logic by the modal operators $\langle\beta\rangle M$. $\mathcal{M}$ is defined by the grammar:

$$M \ ::= \ true \mid M \wedge M \mid \neg M \mid \langle\beta\rangle M \tag{14}$$

The semantics of M (the set of all processes which satisfy M) is defined by relation $\models \subseteq \mathcal{P} \times \mathcal{M}$ where if $(P, M) \in \models$ then we write $P \models M$. Following [HM85], $\models$ is defined as the least set such that:

$$\begin{array}{ll} P \models true & \\ P \models M \wedge N & \text{iff } \ P \models M \ \wedge \ P \models N \\ P \models \neg M & \text{iff } \ \text{not } P \models M \\ P \models \langle\beta\rangle M & \text{iff } \ \exists_{P',s} \ \ P \xrightarrow{s} P' \ \wedge \ \widehat{s} = \beta \ \wedge \ P' \models M \end{array} \tag{15}$$

We abbreviate $\neg true$ as $false$, $\neg\langle\beta\rangle\neg M$ as $[\beta]M$ and for $m > 1$ define $\langle\beta\rangle^m M$ as $\langle\beta\rangle\langle\beta\rangle^{m-1} M$ and $\langle\beta\rangle^1 M$ as $\langle\beta\rangle M$ ($[\beta]^m M$ is defined alike).

Algebraically, we can identify a process with its equivalence class. However, given a logic where properties of processes can be stated and verified, we can identify a process with its properties. When both algebraic and logical views agree, that is when two processes are equivalent iff they have the same properties, then we say that the equivalence is characterised by the logic. Following [HM85], if $P \approx Q$ then P and Q satisfy the same formulas $M \in \mathcal{M}$ and the other way round but only for weak-image-finite $\longrightarrow$:

$$P \approx Q \ \text{ iff } \ \forall_{M \in \mathcal{M}} \ P \models M \Leftrightarrow Q \models M \tag{16}$$

The aim of this section is to provide similar statements for fault-tolerant and conditional fault-tolerant extensions of $\approx$. Consider the new relation $\Vdash \subseteq \mathcal{P} \times \mathcal{M}$ which is defined like $\models$ except that the transitions $\longmapsto$ are now used to define the semantics of formulas $\langle\beta\rangle M$:

$$Q \Vdash \langle\beta\rangle M \ \text{ iff } \ \exists_{Q',s} \ \ Q \stackrel{s}{\longmapsto} Q' \ \wedge \ \widehat{s} = \beta \ \wedge \ Q' \Vdash M \tag{17}$$

Applying $\models$ for the high-level process and $\Vdash$ for the low-level one, we can show that for weak-image-finite relations $\longmapsto$, must-bisimilarity $\sqsubseteq$ is characterised by the Hennessy-Milner logic:

Proposition 1. For weak-image-finite relation $\longmapsto$ we have:

$$P \sqsubseteq Q \ \text{ iff } \ \forall_{M \in \mathcal{M}} \ \ P \models M \ \Leftrightarrow \ Q \Vdash M$$

Proof. ($\Rightarrow$) By induction on the structure of M. ($\Leftarrow$) We show that the relation $B =_{def} \{(P, Q) \mid \forall_{M \in \mathcal{M}} \ \ P \models M \ \Leftrightarrow \ Q \Vdash M\}$ is a must-bisimulation. For details see Appendix A. $\square$

Because $P \sqsubseteq^{?} Q$ implies $P \sqsubseteq^{!} Q$, we also have $P \models M$ iff $Q \Vdash M$ for any $P \sqsubseteq^{?} Q$ and $M \in \mathcal{M}$. The inverse however does not hold, as demonstrated by P and S in Figure 2 which have the same properties (P with respect to $\models$ and S according to $\Vdash$) but still $P \not\sqsubseteq^{?} S$.

Example 3. Consider processes in Figure 2 and two formulas for $m \geq 0$:

- $M = [b]^{2m}\langle a\rangle true$ which asserts that the first action a is always possible after an even number of b's;
- $N = [a]\langle b\rangle^{2m+1}\langle a\rangle true$ where the second occurrence of a may be possible after an odd number of b's.

We have $P \models M \wedge N$ and thus $Q \models M \wedge N$ and $S \models M \wedge N$ because of $P \approx Q \approx S$. We also have $P \sqsubseteq^{!} R$ what gives $R \Vdash M \wedge N$ and $P \sqsubseteq^{!} S$ what results in $S \Vdash M \wedge N$. Finally $Q \Vdash \neg M \wedge N$ and $R \models M \wedge \neg N$. □

Consider $n \in \mathbb{N} \cup \{\infty\}$ which specifies the maximal number of times transitions $\dashrightarrow$ can occur successively. The logical characterisation of $\sqsubseteq^{!}_{n}$ involves conditional satisfaction relation $\Vdash_n$ which is defined in terms of the family of relations $\Vdash^i_n$, indexed by $[0, n]$. Consider $i \in [0, n]$. For all formulas except $\langle\beta\rangle M$, $\Vdash^i_n$ is defined like $\models$. For $\langle\beta\rangle M$ we have:

$$Q \Vdash^i_n \langle\beta\rangle M \;\text{ iff }\; \exists_{Q',s,j} \;\; Q \overset{s}{\longmapsto}{}^i_j Q' \;\wedge\; \widehat{s} = \beta \;\wedge\; Q' \Vdash^j_n M \tag{18}$$

Thus $Q \Vdash^i_n M$ if Q satisfies M, performing transitions $\longrightarrow$ and $\dashrightarrow$ but the last no more than n times in a row and no more than $n - i$ of them initially. Finally, we define $P \Vdash_n M$ iff $P \Vdash^0_n M$. Given such $\Vdash_n$ and provided that $\longmapsto$ is weak-image-finite, we can prove the following characterisation theorem:

Proposition 2. For weak-image-finite relation $\longmapsto$ we have:

$P \sqsubseteq^{!}_{n} Q$ iff $\forall_{M \in \mathcal{M}} \;\; P \models M \;\Leftrightarrow\; Q \Vdash_n M$

Proof. ($\Rightarrow$) By induction on the structure of M. ($\Leftarrow$) We show that the family $\{B_i\}^n_{i=0}$ of relations $B_i =_{def} \{(P,Q) \mid \forall_{M \in \mathcal{M}} \;\; P \models M \;\Leftrightarrow\; Q \Vdash^i_n M\}$ is a conditional must-bisimulation. For details see Appendix A. □

As a result, because $P \sqsubseteq^{?}_{n} Q$ implies $P \sqsubseteq^{!}_{n} Q$, we have $P \models M$ iff $Q \Vdash_n M$ for all $P \sqsubseteq^{?}_{n} Q$ and $M \in \mathcal{M}$.

Example 4. Consider processes in Figure 4, $m \geq 0$ and the following formulas:

- $M = \langle a\rangle true \wedge [a][a] false$
  where action a is possible but then it cannot be followed by another a;
- $N = \langle b\rangle true \wedge [b][a] false$
  where action b is possible but not followed by a either.

We have $P \models [a][b]^{2m+1}(M \wedge N)$ and thus $Q \models [a][b]^{2m+1}(M \wedge N)$ because of $P \approx Q$. For Q we have $Q \Vdash [a][b]^{2m+1}N$ and $Q \Vdash \neg[a][b]^{2m+1}M$, however $Q \Vdash_1 [a][b]^{2m+1}(M \wedge N)$ because then transition $Q''' \overset{a}{\dashrightarrow} Q$, which enables two subsequent actions a, cannot be chosen. For R we have $R \Vdash \neg[a][b]^{2m+1}M$ and $R \Vdash \neg[a][b]^{2m+1}N$, however $R \Vdash_2 [a][b]^{2m+1}(M \wedge N)$ because then transition $R''' \overset{\tau}{\dashrightarrow} R''$ cannot be chosen. □

5 Language of Processes

The structure of a process P has been ignored so far, it was defined as an element of the abstract set $\mathcal{P}$. The more complex is the behaviour of P however, the greater is the need to treat P structurally. In this section, following [Mil89], we define $\mathcal{P}$ as the language of processes which is given the structured operational semantics [Plo81] in terms of the labelled transition system $(\mathcal{P}, \mathcal{A}, \longrightarrow)$.

The syntax of $\mathcal{P}$ is based on two sets of symbols, $\mathcal{A}$ of actions and $\mathcal{X}$ of process identifiers, and involves two syntactic categories, $\mathcal{E}$ of process expressions and $\mathcal{D}$ of declarations. Let $X, Y \in \mathcal{X}$, $L \subseteq \mathcal{L}$ and f be a function over $\mathcal{A}$ such that $f(\tau) = \tau$, $f(a) \neq \tau$ and $f(\overline{a}) = \overline{f(a)}$. $\mathcal{E}$ is defined by the grammar:

$$E \quad ::= \quad X \mid \mathbf{0} \mid \alpha.E \mid E+E \mid E|E \mid E\backslash L \mid E[f] \tag{19}$$

Informally, $\mathbf{0}$ is unable to take any action and $\alpha.E$ performs α and then behaves like E. The operator $+$ represents summation, $|$ parallel composition, $\backslash$ restriction and $[\,]$ renaming. One derived operator is $E \frown F$ where E and F proceed in parallel with actions *out* of E and *in* of F 'joined' and restricted, $E \frown F =_{def} (E[mid/out]|F[mid/in])\backslash\{mid\}$ where mid is not used by E or F.

We use $\mathcal{X}(E)$ for the set of all identifiers in E and $E\{F/X\}$ for the process expression E where all identifiers X are replaced by F. In order to interpret $X \in \mathcal{X}(E)$, we use declarations of the form $X \mathrel{\widehat{=}} F$. If also $X \in \mathcal{X}(F)$ then such X is defined by recursion. Given $X \mathrel{\widehat{=}} F$ and $Y \mathrel{\widehat{=}} G$ where $X \in \mathcal{X}(G)$ and $Y \in \mathcal{X}(F)$, such X and Y are defined by the mutual recursion. In the sequel we will often need to manipulate declarations for mutually recursive identifiers. Then, it will be helpful to use a simple language $\mathcal{D}$ for specifying collections of such declarations. $\mathcal{D}$, ranged over by Δ and ∇, is defined by the grammar:

$$\Delta \quad ::= \quad [\,] \mid \Delta[X \mathrel{\widehat{=}} E] \mid \alpha \odot \Delta \mid \Delta \oplus \nabla \tag{20}$$

Informally, $[\,]$ is an empty declaration and $\Delta[X \mathrel{\widehat{=}} E]$ declares X as E and other identifiers as in Δ. Moreover, $\alpha \odot \Delta$ and $\Delta \oplus \nabla$ perform respective operations (α-prefix and summation) on the right sides of all the corresponding declarations in Δ and ∇. Formally, $\Delta \in \mathcal{D}$ is assigned a partial function $[\![\Delta]\!]$ from $\mathcal{X}$ to $\mathcal{E}$ which is defined in Figure 5 by induction on the structure of Δ. We use $dom(\Delta)$ as the domain of $[\![\Delta]\!]$ ($dom([\,]) =_{def} \emptyset$) and $ran(\Delta)$ as its range.

$$\begin{array}{rll}
[\![\alpha \odot \Delta]\!](X) =_{def} & \alpha.[\![\Delta]\!](X) & \text{if } X \in dom(\Delta) \\
[\![\Delta[Y \mathrel{\widehat{=}} E]]\!](X) =_{def} & \begin{cases} E \\ [\![\Delta]\!](X) \end{cases} & \begin{array}{l} \text{if } X = Y \\ \text{if } X \neq Y,\ X \in dom(\Delta) \end{array} \\
[\![\Delta \oplus \nabla]\!](X) =_{def} & \begin{cases} [\![\Delta]\!](X) \\ [\![\Delta]\!](X) + [\![\nabla]\!](X) \\ [\![\nabla]\!](X) \end{cases} & \begin{array}{l} \text{if } X \in dom(\Delta) - dom(\nabla) \\ \text{if } X \in dom(\Delta) \cap dom(\nabla) \\ \text{if } X \in dom(\nabla) - dom(\Delta) \end{array}
\end{array}$$

Fig. 5. Denotational semantics of declarations $\mathcal{D}$.

We abbreviate $[][X \mathrel{\widehat{=}} E][Y \mathrel{\widehat{=}} F]$ as $[X \mathrel{\widehat{=}} E, Y \mathrel{\widehat{=}} F]$ and write $[X \mathrel{\widehat{=}} E \mid p]$ for all declarations $X \mathrel{\widehat{=}} E$ such that the predicate p holds. Δ is said to be closed if all identifiers in the right side expressions of Δ are declared in Δ:

$$\bigcup_{F \in ran(\Delta)} \mathcal{X}(F) \subseteq dom(\Delta) \tag{21}$$

A process $P \in \mathcal{P}$ is finally the pair $\langle E, \Delta \rangle$ of the process expression E and the closed declaration Δ for all identifiers of E, $\mathcal{X}(E) \subseteq dom(\Delta)$. We write $\langle E, \Delta \rangle \equiv \langle F, \nabla \rangle$ if $\langle E, \Delta \rangle$ and $\langle F, \nabla \rangle$ are identical.

The semantics of $\langle E, \Delta \rangle \in \mathcal{P}$ is defined in terms of the labelled transition system $(\mathcal{P}, \mathcal{A}, \longrightarrow)$ by induction on the structure of E. If $E = X$ then transitions of $\langle X, \Delta \rangle$ involve the semantics of Δ, they are inferred from the transitions of $\langle [\![\Delta]\!](X), \Delta \rangle$. Following [Mil89], transition relation $\longrightarrow$ is the least set defined by inference rules in Figure 6.

$$\frac{}{\langle \alpha.E, \Delta \rangle \xrightarrow{\alpha} \langle E, \Delta \rangle} \qquad \frac{\langle E, \Delta \rangle \xrightarrow{\alpha} \langle E', \Delta \rangle}{\langle E{+}F, \Delta \rangle \xrightarrow{\alpha} \langle E', \Delta \rangle} \qquad \frac{\langle F, \Delta \rangle \xrightarrow{\alpha} \langle F', \Delta \rangle}{\langle E{+}F, \Delta \rangle \xrightarrow{\alpha} \langle F', \Delta \rangle}$$

$$\frac{\langle E, \Delta \rangle \xrightarrow{\alpha} \langle E', \Delta \rangle}{\langle E|F, \Delta \rangle \xrightarrow{\alpha} \langle E'|F, \Delta \rangle} \qquad \frac{\langle F, \Delta \rangle \xrightarrow{\alpha} \langle F', \Delta \rangle}{\langle E|F, \Delta \rangle \xrightarrow{\alpha} \langle E|F', \Delta \rangle}$$

$$\frac{\langle E, \Delta \rangle \xrightarrow{a} \langle E', \Delta \rangle \quad \langle F, \Delta \rangle \xrightarrow{\overline{a}} \langle F', \Delta \rangle}{\langle E|F, \Delta \rangle \xrightarrow{\tau} \langle E'|F', \Delta \rangle}$$

$$\frac{\langle E, \Delta \rangle \xrightarrow{\alpha} \langle E', \Delta \rangle}{\langle E \setminus L, \Delta \rangle \xrightarrow{\alpha} \langle E' \setminus L, \Delta \rangle}, \ \alpha, \overline{\alpha} \notin L \qquad \frac{\langle E, \Delta \rangle \xrightarrow{\alpha} \langle E', \Delta \rangle}{\langle E[f], \Delta \rangle \xrightarrow{f(\alpha)} \langle E'[f], \Delta \rangle}$$

$$\frac{\langle [\![\Delta]\!](X), \Delta \rangle \xrightarrow{\alpha} \langle E, \Delta \rangle}{\langle X, \Delta \rangle \xrightarrow{\alpha} \langle E, \Delta \rangle}, \ X \in dom(\Delta)$$

Fig. 6. Operational semantics of processes $\mathcal{P}$

Example 5. Consider $n \in \mathbb{N} \cup \{\infty\}$ and the process R_n which performs actions a and b; the first at any time; the second no more than n times in a row. We have $R_n =_{def} \langle Y_0, \Delta_n \oplus \nabla_n \rangle$ where:

$$\begin{aligned} \Delta_n &=_{def} [Y_i \mathrel{\widehat{=}} a.Y_0 \quad \mid 0 \le i \le n] \\ \nabla_n &=_{def} [Y_i \mathrel{\widehat{=}} b.Y_{i+1} \mid 0 \le i < n] \end{aligned}$$

□

We compose processes by composing their expressions, using the process combinators (19). For binary operators $+$ and $|$ we assume that the component processes have disjoint sets of identifiers. If $dom(\Delta) \cap dom(\nabla) = \emptyset$ then:

$$\begin{aligned} \alpha.\langle E, \Delta \rangle &=_{def} \langle \alpha.E, \Delta \rangle \\ \langle E, \Delta \rangle \setminus L &=_{def} \langle E \setminus L, \Delta \rangle \\ \langle E, \Delta \rangle[f] &=_{def} \langle E[f], \Delta \rangle \\ \langle E, \Delta \rangle + \langle F, \nabla \rangle &=_{def} \langle E + F, \Delta \oplus \nabla \rangle \\ \langle E, \Delta \rangle \mid \langle F, \nabla \rangle &=_{def} \langle E \mid F, \Delta \oplus \nabla \rangle \end{aligned} \tag{22}$$

In the language defined so far, processes interact by synchronising on complementary actions a and $\overline{a}$. There is no directionality or value which passes between them. For pragmatic reasons, we also need a value-passing language for the set V of values (we assume, for simplicity, that V is finite). To this end we introduce value constants (like ε), value variables (like x and s), value and boolean expressions (like e and p respectively), built using constants, variables and any function symbols we need. The last include $\sharp s$ as the length of the sequence s, s_0 its first element, s' all but the first element and $s : x$ as the sequence s with value x appended. We also introduce parameters into process identifiers: $X(e1, .., en)$ for X of arity n. Then we extend the basic language by input and output prefixes $a(x).E$, $\overline{a}(e).E$ and conditionals **if** p **then** E **else** F. For their translation into the basic language see [Mil89].

Example 6. Consider a buffer Buf_m of capacity $m > 0$, which receives (by action in) and subsequently transmits (by action $\overline{out}$) all values unchanged, in the same order and with at most m of them received but not sent. We have:

$$Buf_m =_{def} \langle X(\varepsilon), \Delta \oplus \nabla \rangle$$

$$\begin{array}{ll} \text{where} & \Delta =_{def} [X(s) \mathrel{\widehat{=}} in(x).X(s:x) \mid 0 \leq \sharp s < m] \\ & \nabla =_{def} [X(s) \mathrel{\widehat{=}} \overline{out}(s_0).X(s') \mid 0 < \sharp s \leq m] \end{array}$$

□

6 Language of Faults

Although a fault is modelled by a set of transitions, using this set directly is not the most convenient way of specifying faults in practice, especially when the abnormal behaviour we want to describe is complex. The purpose of this section is to define the language where faults can be specified and combined.

The idea is to use process identifiers as 'states' which can be affected by faults. Consider a process $\langle X, \Delta \rangle$. Transitions of $\langle X, \Delta \rangle$ can be only inferred from the transitions of $\langle [\![\Delta]\!](X), \Delta \rangle$ where $[\![\Delta]\!](X)$ is the process expression assigned to X by Δ. In order to specify faults, we will use an alternative, 'faulty' declaration $\Psi \in \mathcal{D}$. Suppose that $X \in dom(\Psi)$. Then X is assigned yet another expression $[\![\Psi]\!](X)$ which determines abnormal transitions of $\langle X, \Delta \rangle$, following transitions of $\langle [\![\Psi]\!](X), \Delta \rangle$ and denoted by $\underset{\Psi}{\longmapsto}$:

$$\frac{\langle [\![\Psi]\!](X), \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E, \Delta \rangle}{\langle X, \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E, \Delta \rangle} \tag{23}$$

Transition relation $\underset{\Psi}{\longmapsto}$ is defined as the least set which satisfies inference rules in Figure 7 and used to denote Ψ-affected semantics of $\mathcal{P}$. We also define:

$$Q \overset{\alpha}{\underset{\Psi}{\dashrightarrow}} Q' \text{ iff } Q \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q' \text{ and } Q \overset{\alpha}{\not\longrightarrow} Q' \tag{24}$$

and relations $\underset{\Psi}{\longmapsto}{}^i_j \subseteq \mathcal{P} \times \mathcal{A}^* \times \mathcal{P}$ (10), given $i, j \in [0, n]$ and faulty transitions $\underset{\Psi}{\dashrightarrow}$, specified by Ψ.

$$
\frac{}{\langle \alpha.E, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E, \Delta\rangle}
\qquad
\frac{\langle E, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E', \Delta\rangle}{\langle E{+}F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E', \Delta\rangle}
\qquad
\frac{\langle F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle F', \Delta\rangle}{\langle E{+}F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle F', \Delta\rangle}
$$

$$
\frac{\langle E, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E', \Delta\rangle}{\langle E|F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E'|F, \Delta\rangle}
\qquad
\frac{\langle F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle F', \Delta\rangle}{\langle E|F, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E|F', \Delta\rangle}
$$

$$
\frac{\langle E, \Delta\rangle \xmapsto[\Psi]{a} \langle E', \Delta\rangle \quad \langle F, \Delta\rangle \xmapsto[\Psi]{\overline{a}} \langle F', \Delta\rangle}{\langle E|F, \Delta\rangle \xmapsto[\Psi]{\tau} \langle E'|F', \Delta\rangle}
$$

$$
\frac{\langle E, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E', \Delta\rangle}{\langle E \setminus L, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E' \setminus L, \Delta\rangle},\ \alpha, \overline{\alpha} \notin L
\qquad
\frac{\langle E, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E', \Delta\rangle}{\langle E[f], \Delta\rangle \xmapsto[\Psi]{f(\alpha)} \langle E'[f], \Delta\rangle}
$$

$$
\frac{\langle [\![\Delta]\!](X), \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E, \Delta\rangle}{\langle X, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E, \Delta\rangle},\ X \in dom(\Delta)
\qquad
\frac{\langle [\![\Psi]\!](X), \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E, \Delta\rangle}{\langle X, \Delta\rangle \xmapsto[\Psi]{\alpha} \langle E, \Delta\rangle},\ X \in dom(\Psi)
$$

Fig. 7. Operational semantics of processes $\mathcal{P}$ affected by the fault Ψ.

Ψ is not assumed to be closed. However, in order to ensure that $\xmapsto[\Psi]{}$ does not lead from the well-defined process (where all identifiers are declared) to the ill-defined one, we assume that all process identifiers in the right-side expressions of Ψ are declared by Δ:

$$
\bigcup_{F \in ran(\Psi)} \mathcal{X}(F) \subseteq dom(\Delta) \tag{25}
$$

We use $\mathcal{P}_\Psi \subseteq \mathcal{P}$ for the set of such $\langle E, \Delta\rangle$ and assume that $\xmapsto[\Psi]{} \subseteq \mathcal{P}_\Psi \times \mathcal{A} \times \mathcal{P}_\Psi$ and $\xmapsto[\Psi]{}{}^i_j \subseteq \mathcal{P}_\Psi \times \mathcal{A}^* \times \mathcal{P}_\Psi$.

Example 7. Consider the following declarations which specify various communication faults of the bounded buffer Buf_m, creation (Ψe), corruption (Ψc), omission (Ψo), replication (Ψr) and permutation (Ψp) of messages:

$$
\begin{array}{ll}
\Psi e =_{def} [X(s) \mathrel{\widehat{=}} \tau.\overline{out}(\surd).X(s) & \mid 0 \le \sharp s \le m] \\
\Psi c =_{def} [X(s) \mathrel{\widehat{=}} \tau.\overline{out}(\surd).X(s') & \mid 0 < \sharp s \le m] \\
\Psi o =_{def} [X(s) \mathrel{\widehat{=}} \tau.X(s') & \mid 0 < \sharp s \le m] \\
\Psi r =_{def} [X(s) \mathrel{\widehat{=}} \tau.\overline{out}(s_0).X(s) & \mid 0 < \sharp s \le m] \\
\Psi p =_{def} [X(s) \mathrel{\widehat{=}} \tau.\overline{out}((s')_0).X(s_0 : s'') & \mid 1 < \sharp s \le m]
\end{array}
$$

We use $\surd$ to denote messages which has been corrupted or created. This is a way to abstract from their particular value which is immaterial. Also, we assume that when permuted, only one message is delayed. In order to specify more complex faults, e.g. simultaneous creation, omission and permutation of messages, we can use the summation $\Psi e \oplus \Psi o \oplus \Psi p$ of declarations:

$$
\begin{array}{l}
[X(s) \mathrel{\widehat{=}} \tau.\overline{out}(\surd).X(s) \mid \sharp s = 0] \\
[X(s) \mathrel{\widehat{=}} \tau.\overline{out}(\surd).X(s) + \tau.X(s') \mid \sharp s = 1] \\
[X(s) \mathrel{\widehat{=}} \tau.\overline{out}(\surd).X(s) + \tau.X(s') + \tau.\overline{out}((s')_0).X(s_0 : s'') \mid 1 < \sharp s \le m] \quad \square
\end{array}
$$

7 Fault Transformation of Processes

The primary effect of faults is that a process no longer behaves according to the normal transition relation $\longrightarrow$. In addition to $\longrightarrow$, it can also perform transitions $\underset{\Psi}{\dashrightarrow}$, specified by $\Psi \in \mathcal{D}$. This is a direct, semantic method to represent effects of faults on the behaviour of the process. In this section we present an alternative, syntactic method. The idea is to capture the effect of faults, specified by Ψ, by the process transformation $\mathcal{T}(\,\cdot\,, \Psi)$ where for any $Q \in \mathcal{P}_\Psi$, its behaviour in the Ψ-affected environment is 'the same' as the behaviour of $\mathcal{T}(Q, \Psi)$ in the environment which is free of faults [Liu91, LJ91]. We show that $\mathcal{T}(Q, \Psi)$ yields the binary relation on $\mathcal{P} \times \mathcal{P}_\Psi$ which coincides with $\sqsubseteq$ and the satisfaction relation which agrees with $\models$. In the conditional case we provide a transformation $\widetilde{\mathcal{T}}(\,\cdot\,, \Psi, n)$ which is shown to coincide with $\sqsubseteq_n$ and $\models_n$.

7.1 Fault-Tolerance

Consider $\Psi \in \mathcal{D}$ which specifies transitions $\underset{\Psi}{\dashrightarrow}$ and a process $\langle E, \Delta \rangle \in \mathcal{P}_\Psi$ with the well-defined, Ψ-affected semantics $\underset{\Psi}{\longmapsto}$. If $\langle E, \Delta \rangle$ has no identifiers in common with Ψ then $\langle E, \Delta \rangle$ is not affected by the transitions $\underset{\Psi}{\dashrightarrow}$. Suppose that $X \in dom(\Delta) \cap dom(\Psi)$ and that transitions of $\langle E, \Delta \rangle$ can be inferred from $\langle X, \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta \rangle$. We have either

$$\frac{\langle [\![\Delta]\!](X), \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta \rangle}{\langle X, \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta \rangle} \quad \text{or} \quad \frac{\langle [\![\Psi]\!](X), \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta \rangle}{\langle X, \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta \rangle} \tag{26}$$

where the first transition is normal (it uses Δ to interpret X) and the second is faulty (it uses Ψ). The ability of $\langle E, \Delta \rangle$ to perform the second transition (with respect to $\longrightarrow$) can be syntactically represented by summation, by redefining its process identifier X as $[\![\Delta]\!](X) + [\![\Psi]\!](X)$. To represent the capacity for all transitions in $\underset{\Psi}{\dashrightarrow}$, such a summation must be performed for all identifiers $X \in dom(\Delta) \cap dom(\Psi)$. This leads to the following transformation:

$$\mathcal{T}(\langle E, \Delta \rangle, \Psi) =_{def} \langle E, \Delta \oplus \Psi \rangle \tag{27}$$

We would like to show that $\mathcal{T}(\langle E, \Delta \rangle, \Psi)$ captures the effect of faults, specified by Ψ on $\langle E, \Delta \rangle$. Observe first that declarations (and thus transformation $\mathcal{T}$) 'persist' through the transitions $\underset{\Psi}{\longmapsto}$ and $\longrightarrow$:

$$\begin{array}{l} \text{If } \langle E, \Delta \rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q' \text{ then } Q' \equiv \langle E', \Delta \rangle \text{ for some } E'. \\ \text{If } \mathcal{T}(\langle E, \Delta \rangle, \Psi) \overset{\alpha}{\longrightarrow} R' \text{ then } R' \equiv \mathcal{T}(\langle E', \Delta \rangle, \Psi) \text{ for some } E'. \end{array} \tag{28}$$

Both statements can be shown by transitional induction. Transitional induction is also employed to prove the following lemma (for details see Appendix B):

Lemma 3.

If $\quad Q \in \mathcal{P}_\Psi$
then $Q \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q'$ iff $\mathcal{T}(Q, \Psi) \overset{\alpha}{\longrightarrow} \mathcal{T}(Q', \Psi)$.

Transformation $\mathcal{T}(\,\cdot\,,\Psi)$ induces the satisfaction relation on $\mathcal{P}_\Psi \times \mathcal{M}$ which holds between a process $Q \in \mathcal{P}_\Psi$ and a formula $M \in \mathcal{M}$ iff $\mathcal{T}(Q,\Psi) \models M$. Applying Lemma 3, we can show that this relation coincides with $\VDash$:

Proposition 4.

If $\quad M \in \mathcal{M}$, $Q \in \mathcal{P}_\Psi$ and
$\quad$ transitions $\dashrightarrow$ are specified by Ψ
then $Q \VDash M$ iff $\mathcal{T}(Q,\Psi) \models M$

Proof. By induction on the structure of formulas M, applying Lemma 3. For details see Appendix B. □

Transformation $\mathcal{T}(\,\cdot\,,\Psi)$ also induces the binary relation on $\mathcal{P} \times \mathcal{P}_\Psi$ which holds between $P \in \mathcal{P}$ and $Q \in \mathcal{P}_\Psi$ iff $P \approx \mathcal{T}(Q,\Psi)$. The following proposition asserts that this relation coincides with must-bisimilarity $\sqsubseteq_{!}$:

Proposition 5.

If $\quad P \in \mathcal{P}$, $Q \in \mathcal{P}_\Psi$ and
$\quad$ transitions $\dashrightarrow$ are specified by Ψ
then $P \sqsubseteq_{!} Q \quad$ iff $\quad P \approx \mathcal{T}(Q,\Psi)$

Proof. It is easy to see that for weak-image-finite $\underset{\Psi}{\longmapsto}$, this statement follows from the characterisation theorem (16) and Propositions 1 and 4. For any $\underset{\Psi}{\longmapsto}$, it follows from the fact that:

$$B \subseteq \mathcal{P} \times \mathcal{P}_\Psi \text{ is a must-bisimulation iff } \mathfrak{T}(B,\Psi) \text{ is a bisimulation} \tag{29}$$

where $\mathfrak{T}(B,\Psi) =_{def} \{(P, \mathcal{T}(Q,\Psi)) \mid (P,Q) \in B\}$. For details see Appendix B. □

Thus for must-bisimilarity $\sqsubseteq_{!}$, we have the equivalence diagram (1) of all three approaches to verify fault-tolerance, given $\underset{\Psi}{\dashrightarrow}$ such that $\underset{\Psi}{\longmapsto}$ is weak-image-finite. For $\underset{\Psi}{\longmapsto}$ which is not weak-image-finite, we cannot guarantee that

$P \models M$ iff $Q \VDash M$ for all $M \in \mathcal{M}$ implies $P \sqsubseteq_{!} Q$.

Moreover, for may-bisimilarity $\sqsubseteq_{?}$, we cannot guarantee that $P \approx \mathcal{T}(Q,\Psi)$ implies $P \sqsubseteq_{?} Q$. However, applying equivalences (7) and (29) it is easy to see that $B \subseteq \mathcal{P} \times \mathcal{P}_\Psi$ is a may-bisimulation iff it is a bisimulation together with $\mathfrak{T}(B,\Psi)$. As a result, given transitions $\underset{\Psi}{\dashrightarrow}$, we have the following statements:

$P \approx Q$ iff $(P,Q) \in B$ for B which is a bisimulation
$P \sqsubseteq_{!} Q$ iff $(P,Q) \in B$ for B such that $\mathfrak{T}(B,\Psi)$ is a bisimulation.
$P \sqsubseteq_{?} Q$ iff $(P,Q) \in B$ for B which is a bisimulation together with $\mathfrak{T}(B,\Psi)$.

Example 8. Consider $m, w > 0$ and the task to ensure a reliable communication, specified by the bounded buffer Buf_w, over a medium of capacity m which omits and replicates messages. To this end, we will use a version of the sliding window protocol with the window size w. The protocol consists of two processes, the sender *So* and the receiver *Ro*. The first transmits all messages with their sequence numbers i modulo $w+1$, such that at most w messages are sent without being acknowledged. Suppose, for simplicity, that acknowledgements take place by synchronising *So* and *Ro* on the action *ack*. We use s for the sequence of messages sent but not acknowledged ($\sharp s \leq w$) and repeatedly retransmit s_0. For all arithmetic operations taken modulo $w+1$ we have:

$$
\begin{array}{lllll}
So =_{def} \langle Zs(0,\varepsilon), & [Zs(i,s) & \widehat{=}\ in(x).Zs(i,s,x) & \mid 0 \leq \sharp s < w]\oplus \\
 & [Zs(i,s) & \widehat{=}\ ack.Zs(i,s') & \mid 0 < \sharp s \leq w]\oplus \\
 & [Zs(i,s) & \widehat{=}\ \overline{out}(i-\sharp s, s_0).Zs(i,s) & \mid 0 < \sharp s \leq w] \\
 & [Zs(i,s,x) & \widehat{=}\ \overline{out}(i,x).Zs(i+1, s:x) & \mid 0 \leq \sharp s < w]\oplus \\
 & [Zs(i,s,x) & \widehat{=}\ ack.Zs(i,s',x) & \mid 0 < \sharp s < w]\oplus \\
 & [Zs(i,s,x) & \widehat{=}\ \overline{out}(i-1,s_0).Zs(i,s,x) & \mid 0 < \sharp s < w]\rangle \\
Ro =_{def} \langle Zr(0), & [Zr(i) & \widehat{=}\ in(j,x).\ \mathbf{if}\ i=j & \\
 & & \qquad \mathbf{then}\ \overline{out}(x).\overline{ack}.Zr(i+1) & \\
 & & \qquad \mathbf{else}\ Zr(i)]\rangle &
\end{array}
$$

Given such *So* and *Ro*, we can prove that:

$$Buf_w \approx (So \frown \mathcal{T}(Buf_m, \Psi o \oplus \Psi r) \frown Ro) \setminus \{ack\}$$

and consequently $Buf_w \sqsubseteq (So \frown Buf_m \frown Ro) \setminus \{ack\}$, following Proposition 5 and the fact that processes *So* and *Ro* are not affected by $\Psi o \oplus \Psi r$ (they have disjoint sets of identifiers). However, the above statement does not depend on the transitions specified by $\Psi o \oplus \Psi r$ and we can find a bisimulation B such that $(Buf_w, (So \frown Buf_m \frown Ro) \setminus \{ack\}) \in B$ and $\mathfrak{T}(B, \Psi o \oplus \Psi r)$ is a bisimulation. Thus we have $Buf_w \sqsubseteq (So \frown Buf_m \frown Ro) \setminus \{ack\}$. □

7.2 Conditional Fault-Tolerance

Consider declaration Ψ and a process $\langle E, \Delta\rangle \in \mathcal{P}_\Psi$. Suppose that we want to verify $\langle E, \Delta\rangle$ in the presence of transitions $\overset{}{\underset{\Psi}{\dashrightarrow}}$ and under assumption n about their quantity. To this end, like before, we will use process transformations. The idea is to use a family $\{X_i\}_{i=0}^n$ of identifiers for each identifier X of Δ or Ψ. Consider $i \in [0, n]$ and the following transformations $\widehat{\mathcal{T}}_i(\cdot, n)$ and $\widetilde{\mathcal{T}}_i(\cdot, \Psi, n)$:

$$
\begin{array}{l}
\widehat{\mathcal{T}}_i(\langle E, \Delta\rangle, n) \quad\ =_{def} \langle E_i, \Delta_n\rangle \quad \text{and} \\
\widetilde{\mathcal{T}}_i(\langle E, \Delta\rangle, \Psi, n) =_{def} \mathcal{T}(\widehat{\mathcal{T}}_i(\langle E, \Delta\rangle, n), \Psi_n)
\end{array}
$$

$$
\begin{array}{lll}
\text{where} & E_i & =_{def} E\{X_i/X \mid X \in \mathcal{X}(E)\} \\
 & \Delta_n & =_{def} [X_i \,\widehat{=}\, F\{Y_0/Y \mid Y \in \mathcal{X}(F)\} \mid [\![\Delta]\!](X) = F \wedge 0 \leq i \leq n] \\
 & \Psi_n & =_{def} [X_i \,\widehat{=}\, F\{Y_{i+1}/Y \mid Y \in \mathcal{X}(F)\} \mid [\![\Psi]\!](X) = F \wedge 0 \leq i < n]
\end{array}
$$

Thus $\widetilde{\mathcal{T}}_i$ is defined in terms of transformations $\mathcal{T}$ and $\widehat{\mathcal{T}}_i$. Also, Δ_n is obtained from Δ by replacing each declaration $X \mathrel{\widehat{=}} F$ with the family of declarations $X_i \mathrel{\widehat{=}} F\{Y_0/Y \mid Y \in \mathcal{X}(F)\}$, for all $i \in [0,n]$, and similarly for Ψ_n but only given $i \in [0,n)$. Finally, we define $\widehat{\mathcal{T}}(\cdot,n)$ and $\widetilde{\mathcal{T}}(\cdot,\Psi,n)$ taking $i=0$:

$$\begin{aligned} \widehat{\mathcal{T}}(\langle E,\Delta\rangle,n) &=_{def} \widehat{\mathcal{T}}_0(\langle E,\Delta\rangle,n) \\ \widetilde{\mathcal{T}}(\langle E,\Delta\rangle,\Psi,n) &=_{def} \widetilde{\mathcal{T}}_0(\langle E,\Delta\rangle,\Psi,n) \end{aligned} \tag{30}$$

Recall the family $\{\overset{}{\underset{\Psi}{\longmapsto}}{}^i_j\}_{i,j=0}^n$ of relations which denotes the effect of transitions $\underset{\Psi}{\dashrightarrow}$ on the semantics of $\mathcal{P}_\Psi$, under assumption n about their quantity. There are two problems to obtain the same effect using transformations:

1. Consider $X \in dom(\Delta) \cap dom(\Psi)$ and $\langle X,\Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E,\Delta\rangle$ which can be inferred from $\langle [\![\Delta]\!](X),\Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E,\Delta\rangle$ or $\langle [\![\Psi]\!](X),\Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E,\Delta\rangle$. The problem appears when $\langle X,\Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E,\Delta\rangle$ can be inferred from both of them. Then it is always regarded as 'normal' by $\underset{\Psi}{\longmapsto}{}^i_j$ but not always by $\widetilde{\mathcal{T}}(\cdot,\Psi,n)$. If no such E and α exists then we say that *Ψ has the proper effect on Δ*.
2. The second problem is that in case of $\widetilde{\mathcal{T}}(\cdot,\Psi,n)$ (but not $\underset{\Psi}{\longmapsto}{}^i_j$), some transitions do not 'update' the index i. Suppose that $\Psi =_{def} [X \mathrel{\widehat{=}} \tau.a.X]$ and $\Delta =_{def} [X \mathrel{\widehat{=}} b.X]$. Then we have: $\langle X,\Delta\rangle \overset{\tau}{\underset{\Psi}{\dashrightarrow}} \langle a.X,\Delta\rangle \overset{a}{\longrightarrow} \langle X,\Delta\rangle$ and thus $\langle X,\Delta\rangle \overset{\tau a}{\underset{\Psi}{\longmapsto}}{}^0_0 \langle X,\Delta\rangle$, however $\widetilde{\mathcal{T}}_0(\langle X,\Delta\rangle,\Psi,n) \overset{\tau a}{\longrightarrow} \widetilde{\mathcal{T}}_1(\langle X,\Delta\rangle,\Psi,n)$. We can solve this problem assuming that all expressions involved are *linear* i.e. they are of the form $\sum_{i=1}^k \alpha_i.X_i$ (Δ is linear if all $F \in ran(\Delta)$ are linear).

Under assumption about the linear form of Δ and Ψ, we can easily show that:

$$\begin{aligned} &\text{if } \langle X,\Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q' \text{ then } Q' \equiv \langle Y,\Delta\rangle \text{ and} \\ &\text{if } \widetilde{\mathcal{T}}_i(\langle X,\Delta\rangle,\Psi,n) \overset{\alpha}{\longrightarrow} R' \text{ then } R' \equiv \widetilde{\mathcal{T}}_j(\langle Y,\Delta\rangle,\Psi,n) \end{aligned} \tag{31}$$

where $Y \in \mathcal{X}$ and $j \in [0,n]$. Then we have the following lemma:

Lemma 6.

If $\quad \langle X,\Delta\rangle \in \mathcal{P}_\Psi$ where
$\quad\quad \Psi$ has the proper effect on Δ and Ψ and Δ are linear
then $\langle X,\Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^i_j \langle Y,\Delta\rangle$ iff $\widetilde{\mathcal{T}}_i(\langle X,\Delta\rangle,\Psi,n) \overset{s}{\longrightarrow} \widetilde{\mathcal{T}}_j(\langle Y,\Delta\rangle,\Psi,n)$

Proof. By induction on the length of s. For details see Appendix C. □

Like before, transformation $\widetilde{\mathcal{T}}(\,\cdot\,,\Psi,n)$ induces two relations: the satisfaction relation which holds between $Q \in \mathcal{P}_\Psi$ and $M \in \mathcal{M}$ iff $\widetilde{\mathcal{T}}(Q,\Psi,n) \models M$ and the binary relation which holds between $P \in \mathcal{P}$ and $Q \in \mathcal{P}_\Psi$ iff $P \approx \widetilde{\mathcal{T}}(Q,\Psi,n)$. Under assumptions of Lemma 6, we can show that the first relation coincides with conditional satisfaction relation and the second with conditional must-bisimilarity:

Proposition 7.

If $\quad M \in \mathcal{M}$ and $\langle X,\Delta\rangle \in \mathcal{P}_\Psi$ where
$\quad\quad \Psi$ has the proper effect on Δ and Ψ and Δ are linear
then $\langle X,\Delta\rangle \models_n M$ iff $\widetilde{\mathcal{T}}(\langle X,\Delta\rangle,\Psi,n) \models M$

Proof. We show that for all $i \in [0, n]$, $\langle X, \Delta \rangle \Vdash_n^i M$ iff $\widetilde{\mathcal{T}}_i(\langle X, \Delta \rangle, \Psi, n) \models M$. The proof proceeds by induction on the structure of M, applying Lemma 6. For details see Appendix C. $\square$

Proposition 8.

If $\quad P \in \mathcal{P}$ and $\langle X, \Delta \rangle \in \mathcal{P}_\Psi$ where
Ψ has the proper effect on Δ and Ψ and Δ are linear
then $P \sqsubseteq_n \langle X, \Delta \rangle$ iff $P \approx \widetilde{\mathcal{T}}(\langle X, \Delta \rangle, \Psi, n)$

Proof. For weak-image-finite $\underset{\Psi}{\longmapsto}$, this statement follows from (16) and Propositions 2 and 7. For any $\underset{\Psi}{\longmapsto}$ and family $\{B_i\}_{i=0}^n$ such that if $(P, Q) \in B_i$ then $Q \equiv \langle X, \Delta \rangle$, it follows from the fact that:

$$\begin{array}{l} \{B_i\}_{i=0}^n \text{ is a conditional must-bisimulation iff} \\ \widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n) \text{ is a bisimulation} \end{array} \tag{32}$$

where $\widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n) =_{def} \bigcup_{i=0}^n \{(P, \widetilde{\mathcal{T}}_i(Q, \Psi, n)) \mid (P, Q) \in B_i\}$.
For details of the proof see Appendix C. $\square$

Assuming additionally that all right-side expressions in Ψ are of the form $\sum_{i=1}^k \tau.X_i$, the same result can be obtained applying auxiliary actions, concurrent composition and processes R_n (Example 5):

$$\widetilde{\mathcal{T}}'(\langle X, \Delta \rangle, \Psi, n) =_{def} (\langle X, (\Delta \odot \overline{a}) \oplus (\overline{b} \odot \Psi) \rangle \mid R_n) \backslash \{a, b\}$$

where $[\![\Delta \odot \overline{a}]\!](X) = [\![\Delta]\!](X)\{\overline{a}.Y/Y \mid Y \in \mathcal{X}([\![\Delta]\!](X))\}$.

Thus for $\sqsubseteq_n$ and under assumptions of Lemma 6, we have the equivalence diagram similar to (1), given $\underset{\Psi}{\dashrightarrow}$ such that $\underset{\Psi}{\longmapsto}$ is weak-image-finite. Consider $\widehat{\mathfrak{T}}(\{B_i\}_{i=0}^n, n) =_{def} \bigcup_{i=0}^n \{(P, \widehat{\mathcal{T}}_i(Q, n)) \mid (P, Q) \in B_i\}$. For $\sqsubseteq_n$ we can show that $\{B_i\}_{i=0}^n$ is a conditional may-bisimulation iff $\widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n)$ and $\widehat{\mathfrak{T}}(\{B_i\}_{i=0}^n, n)$ are bisimulations.

Although the linear form of Δ and Ψ is necessary to establish these results, the meaning of $\widetilde{\mathcal{T}}$ and $\widehat{\mathcal{T}}$ for non-linear Δ and Ψ is also well-understood. While in the first case all transitions are significant, they all 'update' the index i, in the second case only chosen ones are significant. The main reason for 'mismatch' between $\widetilde{\mathcal{T}}(\cdot, \Psi, n)$ and $\underset{\Psi}{\longmapsto}{}_j^i$ for non-linear expressions lies in the restrictive form of the latter. In the following example we will illustrate using transformations $\widetilde{\mathcal{T}}$ to verify conditional fault-tolerance for $\langle E, \Delta \rangle$ and Ψ which is not linear.

Example 9. Consider $n, m > 0$ and the task to ensure a reliable communication (specified by Buf_{m+n+2}) over a medium of capacity m which permutes messages. To this end we will use two processes: the sender Sp_n and the receiver Rp_n. In order to determine the proper transmission order, messages will be send by Sp_n with their sequence numbers modulo n. The value of n determines the number of parallel components St_i of Rp_n ($i = 0, \ldots, n-1$), each one used to store a message with the sequence value i, received out-of-order. The value of $\perp$ means that no message is stored. Suppose that the summation $i+1$ below is taken modulo n. Then we have:

$$
\begin{array}{lll}
Sp_n & =_{def} & \langle Zs(0), [Zs(i) \mathrel{\widehat{=}} in(x).\overline{out}(x,i).Zs(i+1) \mid 0 \le i < n]\rangle \\
Rp_n & =_{def} & (Ctr \mid St_0 \mid \cdots \mid St_{n-1}) \backslash \{st_0, \ldots, st_{n-1}\} \\
Ctr & =_{def} & \langle Zr(0),\ [Zr(i) \ \mathrel{\widehat{=}}\ in(x,j).\ \mathbf{if}\ i = j \\
& & \qquad\qquad\qquad\qquad\qquad\quad \mathbf{then}\ \overline{out}(x).Zm(i+1) \\
& & \qquad\qquad\qquad\qquad\qquad\quad \mathbf{else}\ \ \overline{st_j}(x).Zr(i) \qquad \mid 0 \le i < n] \\
& & \qquad\quad [Zm(i) \mathrel{\widehat{=}}\ st_i(x).\ \ \mathbf{if}\ x = \bot \\
& & \qquad\qquad\qquad\qquad\qquad\quad \mathbf{then}\ Zr(i) \\
& & \qquad\qquad\qquad\qquad\qquad\quad \mathbf{else}\ \ \overline{out}(x).Zm(i+1) \mid 0 \le i < n]\rangle \\
St_i & =_{def} & \langle Zm_i,\ \ [Zm_i \ \mathrel{\widehat{=}}\ st_i(x).\overline{st_i}(x).Zm_i + \overline{st_i}(\bot).Zm_i]\rangle,\ \ 0 \le i < n
\end{array}
$$

We can shown that the process $Sp_n \frown Buf_m \frown Rp_n$ tolerates Ψp, provided the number of successive permutations is not greater than n:

$$
Buf_{m+n+2} \approx \widetilde{T}(Sp_n \frown Buf_m \frown Rp_n, \Psi p, n) \qquad \square
$$

8 Conclusions

Currently, there is a number of methods for specifying and proving correctness of systems which are tolerant of faults in the operating environment [Cri85, JH87, LJ91, Nor92, Pel91, PJ93, Pra87]. Based on different formalisms and various semantic models, of systems and faults, using different ways to represent effects of faults on the behaviour of systems, they are difficult to compare and relate. In particular, it is not certain whether a system which is fault-tolerant with respect to one of these methods is also fault-tolerant according to the others.

This relationship is clear for three methods defined in this paper: algebraic, logical and transformational. Based on the common semantic model of labelled transition systems, which is also used to model faults, all three methods have been proved equivalent for certain classes of faults. The equivalence holds in two cases, unconditional, where no assumption is made about the quantity of faults, and conditional, given the maximal number of times they can occur successively.

There is a number of directions that we plan to develop this work. We plan to study the use of other bisimulation-like relations, like the partial [Wal90], the 'terminating' [AH92] and the context-dependent [Lar87] bisimilarities for fault-tolerance. In the presence of faulty transitions, a convergent process may diverge and the one which terminates (successfully) may deadlock. We plan to relate our theory with modal specifications [LT88] which constrain possible implementations by two kinds of transitions, necessary and admissible (any necessary transition is also admissible). Bisimulation gives rise to the refinement ordering between modal specifications which is different however from the relations defined in this paper. We plan to determine the class of contexts (built from the operators of the process language) where our relations, especially the stronger, may-bisimilarity, is substitutive. Last but not least, we plan to support the development of fault-tolerant processes, based on the verification theory of this paper and using the decomposition of faults specified in our language.

Acknowledgements

I am grateful to my supervisor, Mathai Joseph, for many valuable comments on draft versions of this paper, to Zhiming Liu for stimulating discussions on fault-tolerance, and to David Walker for his reading, helpful comments and for putting some literature to my attention.

References

[AH92] L. Aceto and M. Hennessy. Termination, deadlock and divergence. *Journal of ACM*, 39(1):147–187, 1992.

[Cri85] F. Cristian. A rigorous approach to fault-tolerant programming. *IEEE Transactions on Software Engineering*, 11(1):23–31, 1985.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32(1):137–161, 1985.

[JH87] He Jifeng and C.A.R. Hoare. Algebraic specification and proof of a distributed recovery algorithm. *Distributed Computing*, 2:1–12, 1987.

[Kel76] R. Keller. Formal verification of parallel programs. *Communications of ACM*, 19(7):561–572, 1976.

[Lar87] K.G. Larsen. A context dependent equivalence between processes. *Theoretical Computer Science*, 49:185–215, 1987.

[Liu91] Z. Liu. *Fault-Tolerant Programming by Transformations*. PhD thesis, University of Warwick, 1991.

[LJ91] Z. Liu and M. Joseph. Transformations of programs for fault-tolerance. *Formal Aspects of Computing*, 4:442–469, 1991.

[LT87] N. Lynch and M. Tuttle. Hierarchical correctness proofs for distributed algorithms. Technical report, MIT Laboratory for Computer Science, 87.

[LT88] K.G. Larsen and B. Thomsen. A modal process logic. In *Proc. 3rd Annual Symposium on Logic in Computer Science*, pages 203–210, 88.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall International, 1989.

[MP91] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*, volume 1. Springer-Verlag, 1991.

[Nor92] J. Nordahl. *Specification and Design of Dependable Communicating Systems*. PhD thesis, Technical University of Denmark, 1992.

[Par81] D. Park. Concurrency and automata on infinite sequences. *LNCS*, 104, 81.

[Pel91] J. Peleska. Design and verification of fault tolerant systems with CSP. *Distributed Computing*, 5:95–106, 1991.

[PJ93] D. Peled and M. Joseph. A compositional approach for fault-tolerance using specification transformation. *LNCS*, 694, 1993.

[Plo81] G. Plotkin. A structural approach to operational semantics. Technical report, Computer Science Department, Aarhus University, 81.

[Pra86] V. Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1):33–71, 1986.

[Pra87] K.V.S. Prasad. *Combinators and Bisimulation Proofs for Restartable Systems*. PhD thesis, Department of Computer Science, University of Edinburgh, 1987.

[Wal90] D.J. Walker. Bisimulation and divergence. *Information and Computation*, 85:202–241, 90.

[Win89] G. Winskel. An introduction to event structures. *LNCS*, 354:364–397, 1989.

A Proofs from Section 4

Lemma 9.
B is a must-bisimulation iff for all $(P,Q) \in B$ and $s \in \mathcal{A}^*$:

$$\text{whenever } P \xrightarrow{s} P' \text{ then } \exists_{Q',t}\ Q \xmapsto{t} Q' \wedge \widehat{s} = \widehat{t} \wedge (P',Q') \in B$$
$$\text{whenever } Q \xmapsto{s} Q' \text{ then } \exists_{P',t}\ P \xrightarrow{t} P' \wedge \widehat{s} = \widehat{t} \wedge (P',Q') \in B$$

Proof. ($\Rightarrow$) Consider $(P,Q) \in B$ and let $P \equiv P_0 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} P_n \equiv P'$ where $n \geq 0$ and $s = \alpha_1 \ldots \alpha_n$. If $n = 0$ then $P' \equiv P$ and it is enough to take $t = \varepsilon$ and $Q' \equiv Q$. If $n > 0$ then for all $k \in [1,n]$ there exists Q_k and t_k such that $Q_{k-1} \xmapsto{t_k} Q_k$, $\widehat{t_k} = \widehat{\alpha_k}$ and $(P_k, Q_k) \in B$. Let $t =_{def} t_1 : \cdots : t_n$. Then $Q \xmapsto{t} Q'$, $\widehat{t} = \widehat{s}$ and $(P',Q') \in B$. For $Q \xmapsto{s} Q'$ the proof is the same.
($\Leftarrow$) It is enough to take s such that $\sharp s = 1$. □

Proof of Proposition 1.
($\Rightarrow$) Let $P \mathrel{\boxed{!}} Q$ and $M \in \mathcal{M}$. We will show that $P \models M$ iff $Q \mathrel{\|\!\!=} M$, by induction on M. Let $P \models \langle\beta\rangle M_1$ (for $true$, $M_1 \wedge M_2$ and $\neg M_1$ the proof is obvious). Then $P \xrightarrow{s} P'$ where $\widehat{s} = \beta$ and $P' \models M_1$ for some s and P'. Applying Lemma 9, there exists t and Q' such that $Q \xmapsto{t} Q'$, $\widehat{t} = \widehat{s}$ and $P' \mathrel{\boxed{!}} Q'$. Then, by induction we have $Q' \mathrel{\|\!\!=} M_1$ and finally $Q \mathrel{\|\!\!=} \langle\beta\rangle M_1$. For $Q \mathrel{\|\!\!=} \langle\beta\rangle M_1$ the proof is similar.
($\Leftarrow$) Consider $B =_{def} \{(P,Q) \mid \forall_{M \in \mathcal{M}}\ P \models M \Leftrightarrow Q \mathrel{\|\!\!=} M\}$ and suppose that $\longmapsto$ is weak-image-finite. We will show that B is a must-bisimulation. Let $(P,Q) \in B$ and $P \xrightarrow{\alpha} P'$. Then $P \models \langle\widehat{\alpha}\rangle true$ and $Q \mathrel{\|\!\!=} \langle\widehat{\alpha}\rangle true$ what gives $Q \xmapsto{s} Q'$ for some s ($\widehat{s} = \widehat{\alpha}$) and Q'. Let $\mathcal{Q}$ be the set of all such Q'. Because $\longmapsto$ is weak-image-finite, we have $\mathcal{Q} = \{Q_i\}_{i=1}^n$ for some $n > 0$. Then, it is enough to find $i \in [1,n]$ such that $(P', Q_i) \in B$. Suppose on the contrary, that for all $i \in [1,n]$ there exists $M_i \in \mathcal{M}$ such that $P' \models M_i$ and $Q_i \not\mathrel{\|\!\!=} M_i$. Let $M =_{def} M_1 \wedge \cdots \wedge M_n$. Then we have $P' \models M$ and so $P \models \langle\widehat{\alpha}\rangle M$, however $Q \not\mathrel{\|\!\!=} \langle\widehat{\alpha}\rangle M$ in spite of $(P,Q) \in B$. For $Q \xmapsto{\alpha} Q'$ the proof is the same ($\longrightarrow$ is also weak-image-finite). □

Lemma 10.
If $Q \xmapsto{s}{}^i_j Q' \xmapsto{t}{}^j_k Q''$ then $Q \xmapsto{s:t}{}^i_k Q''$.

Proof. By induction on the length of s. If $Q \xmapsto{\varepsilon}{}^i_j Q' \xmapsto{t}{}^j_k Q''$ then we have $Q' \equiv Q$ and $i = j$ what implies $Q \xmapsto{\varepsilon:t}{}^i_k Q''$. If $Q \xmapsto{\alpha:s}{}^i_j Q' \xmapsto{t}{}^j_k$ then either $i < n$ and $Q \overset{\alpha}{\dashrightarrow} Q''' \xmapsto{s}{}^{i+1}_j Q' \xmapsto{t}{}^j_k Q''$ or $Q \xrightarrow{\alpha} Q''' \xmapsto{s}{}^0_j Q' \xmapsto{t}{}^j_k Q''$ for some Q'''. In the first case by induction $Q''' \xmapsto{s:t}{}^{i+1}_k Q''$ so we get $Q \xmapsto{\alpha:s:t}{}^i_k Q''$. In the second case by induction $Q''' \xmapsto{s:t}{}^0_k Q''$ so we finally get $Q \xmapsto{\alpha:s:t}{}^i_k Q''$. □

Lemma 11.
Consider $\{B_i\}_{i=0}^n$ where $B_i \subseteq \mathcal{P} \times \mathcal{P}$ for $i \in [0,n]$. $\{B_i\}_{i=0}^n$ is a conditional must-bisimulation iff for all $i, j \in [0,n]$, $(P,Q) \in B_i$ and $s \in \mathcal{A}^*$:

$$\text{whenever } P \xrightarrow{s} P' \text{ then } \exists_{Q',j,t}\ Q \xmapsto{t}{}^i_j Q' \wedge \widehat{t} = \widehat{s} \wedge (P',Q') \in B_j$$
$$\text{whenever } Q \xmapsto{s}{}^i_j Q' \text{ then } \exists_{P',t}\ \ P \xrightarrow{t} P' \wedge \widehat{t} = \widehat{s} \wedge (P',Q') \in B_j$$

Proof. ($\Rightarrow$) Consider $i \in [0,n]$, $(P,Q) \in B_i$ and $s = \alpha_1 \dots \alpha_n$. Suppose that $P \equiv P_0 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_n} P_n \equiv P'$ for $n \geq 0$. If $n = 0$ then it is enough to take $t = \varepsilon$, $j = i$ and $Q' \equiv Q$. It $n > 0$ then for all $k \in [1,n]$ there exists Q_k, t_k and i_k such that $Q_{k-1} \overset{t_k}{\longmapsto}{}^{i_{k-1}}_{i_k} Q_k$, $\widehat{t_k} = \widehat{\alpha_k}$ and $(P_k, Q_k) \in B_{i_k}$ where $Q_0 \equiv Q$ and $i_0 = i$. Let $t = t_1 : \cdots : t_n$. Then $\widehat{t} = \widehat{s}$, $(P', Q_n) \in B_{i_n}$ and from Lemma 10, $Q \overset{t}{\longmapsto}{}^{i}_{i_n} Q_n$. For $Q \overset{s}{\longmapsto}{}^{i}_{j} Q'$ the proof is similar.
($\Leftarrow$) It is enough to take s such that $\sharp s = 1$. □

Proof of Proposition 2.
($\Rightarrow$) Let $i \in [0,n]$, $P \sqsubseteq^{i}_{n} Q$ and $M \in \mathcal{M}$. We will show that $P \models M$ iff $Q \Vdash^{i}_{n} M$ by induction on M. Let $P \models \langle\beta\rangle M_1$ (for $true$, $M_1 \wedge M_2$ and $\neg M_1$ the proof is obvious). Then $P \xrightarrow{s} P'$ and $\widehat{s} = \beta$ for some P' such that $P' \models M_1$. Applying Lemma 11, there exists t ($\widehat{t} = \beta$) and j such that $Q \overset{t}{\longmapsto}{}^{i}_{j} Q'$ and $P' \sqsubseteq^{j}_{n} Q'$ for some Q'. Then by induction we have $Q' \Vdash^{j}_{n} M_1$ and thus $Q \Vdash^{i}_{n} \langle\beta\rangle M_1$. For $Q \Vdash^{i}_{n} \langle\beta\rangle M_1$ the proof is similar.
($\Leftarrow$) Let $B_i =_{def} \{(P,Q) \mid \forall_{M \in \mathcal{M}}\ P \models M \Leftrightarrow Q \Vdash^{i}_{n} M\}$ and suppose that $\longmapsto$ is weak-image-finite. We will show that $\{B_i\}_{i=0}^{n}$ is a conditional must-bisimulation. Suppose that $i \in [0,n]$, $(P,Q) \in B_i$ and $P \xrightarrow{\alpha} P'$. Then we have $P \models \langle\widehat{\alpha}\rangle true$ and $Q \Vdash^{i}_{n} \langle\widehat{\alpha}\rangle true$ so $Q \overset{s}{\longmapsto}{}^{i}_{j} Q'$ for some j and s such that $\widehat{s} = \widehat{\alpha}$. Let $\mathcal{Q}_j$ be the set of all such Q'. Because $\longmapsto$ is weak-image-finite, we have $\mathcal{Q}_j = \{Q^j_k\}_{k=1}^{k_j}$ where $k_j \geq 0$ and there exists j such that $k_j > 0$. Then it is enough to show that for some j there exists $k \in [1,k_j]$ such that $(P', Q^j_k) \in B_j$. Suppose on the contrary: for all $j \in [0,n]$ and for all $k \in [1,k_j]$ there exists $M^j_k \in \mathcal{M}$ such that $P' \models M^j_k$ and $Q^j_k \not\Vdash M^j_k$. Let $M =_{def} \bigwedge_{j=0}^{n} \bigwedge_{k=1}^{k_j} M^j_k$. Then we have $P' \models M$ so $P \models \langle\widehat{\alpha}\rangle M$, however $Q \not\Vdash^{i}_{n} \langle\widehat{\alpha}\rangle M$ in spite of $(P,Q) \in B_i$. For $Q \overset{s}{\longmapsto}{}^{i}_{j} Q'$ the proof is the same, because $\longrightarrow$ is also weak-image-finite. □

B Proofs from Section 7.1

Proof of Lemma 3.
We proceed by transitional induction. Let $\Gamma =_{def} \Delta \oplus \Psi$.
($\Rightarrow$) If $Q \equiv \langle E, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q'$ then $Q' \equiv \langle E', \Delta\rangle$ (28) and we will show that $\langle E, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$ by induction on the inference of transition $\langle E, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. There are six cases:

1. $E \equiv \alpha E'$. Then $\langle \alpha.E', \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$.
2. $E \equiv E_1 + E_2$. Then either $\langle E_1, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$ or $\langle E_2, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. For the first (the second is symmetrical), by induction $\langle E_1, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$ and so we have $\langle E_1 + E_2, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$.
3. $E \equiv E_1 | E_2$. Then there are three cases:
   (a) $\langle E_1, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E_1', \Delta\rangle$ where $E' \equiv E_1' | E_2$. Then by induction we have $\langle E_1, \Gamma\rangle \xrightarrow{\alpha} \langle E_1', \Gamma\rangle$ and so $\langle E_1 | E_2, \Gamma\rangle \xrightarrow{\alpha} \langle E_1' | E_2, \Gamma\rangle$.
   (b) $\langle E_2, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E_2', \Delta\rangle$ where $E' \equiv E_1 | E_2'$ is similar.
   (c) $\langle E_1, \Delta\rangle \overset{a}{\underset{\Psi}{\longmapsto}} \langle E_1', \Delta\rangle$ and $\langle E_2, \Delta\rangle \overset{\overline{a}}{\underset{\Psi}{\longmapsto}} \langle E_2', \Delta\rangle$ for $\alpha = \tau$ and $E' \equiv E_1' | E_2'$. By induction $\langle E_1, \Gamma\rangle \xrightarrow{a} \langle E_1', \Gamma\rangle$ and $\langle E_2, \Gamma\rangle \xrightarrow{\overline{a}} \langle E_2', \Gamma\rangle$ and so we have $\langle E_1 | E_2, \Gamma\rangle \xrightarrow{\tau} \langle E_1' | E_2', \Gamma\rangle$.

4. $E \equiv E_1 \setminus L$. Then $\langle E_1, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E_1', \Delta\rangle$ for $\alpha, \overline{\alpha} \notin L$ and $E' \equiv E_1' \setminus L$. Thus by induction $\langle E_1, \Gamma\rangle \xrightarrow{\alpha} \langle E_1', \Gamma\rangle$ and so $\langle E_1 \setminus L, \Gamma\rangle \xrightarrow{\alpha} \langle E_1' \setminus L, \Gamma\rangle$.
5. $E \equiv E_1[f]$ is similar.
6. $E \equiv X$. Then either $\langle [\![\Delta]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$ or $\langle [\![\Psi]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. Consider the first (the second is symmetrical), then $X \in dom(\Delta)$ and by induction $\langle [\![\Delta]\!](X), \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$. There are two cases:
   (a) $X \notin dom(\Psi)$. Then $\langle X, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$ because $[\![\Gamma]\!](X) = [\![\Delta]\!](X)$.
   (b) $X \in dom(\Psi)$. Then $[\![\Gamma]\!](X) = [\![\Delta]\!](X) + [\![\Psi]\!](X)$ from (25) and so because $\langle [\![\Delta]\!](X) + [\![\Psi]\!](X), \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$, we have $\langle X, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$.

($\Leftarrow$) Let $\mathcal{T}(Q, \Psi) \xrightarrow{\alpha} R'$ where $Q \equiv \langle E, \Delta\rangle$. Then $R' \equiv \mathcal{T}(\langle E', \Delta\rangle, \Psi)$ applying (28) and we will show that $\langle E, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$, using the similar induction on the inference of the transition $\langle E, \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$. Consider $E \equiv X$ only. Then $\langle [\![\Gamma]\!](X), \Gamma\rangle \xrightarrow{\alpha} \langle E', \Gamma\rangle$ and by induction we have $\langle [\![\Gamma]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. Thus there are two cases:

1. $X \notin dom(\Psi)$. Then $[\![\Gamma]\!](X) = [\![\Delta]\!](X)$ so $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$.
2. $X \in dom(\Psi)$. Then $[\![\Gamma]\!](X) = [\![\Delta]\!](X) + [\![\Psi]\!](X)$ applying (25) so we either have $\langle [\![\Delta]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$ or $\langle [\![\Psi]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. In both cases we finally get $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle E', \Delta\rangle$. □

Proof of Proposition 4.
We will show that $Q \models\!\!\models M$ iff $\mathcal{T}(Q, \Psi) \models M$ by induction on the structure of M. Let $\mathcal{T}(Q, \Psi) \models \langle\beta\rangle M_1$ (for $true$, $M_1 \wedge M_2$ and $\neg M_1$ the proof is obvious). Then $\mathcal{T}(Q, \Psi) \xrightarrow{s} R'$ where $\widehat{s} = \beta$, $R' \models M_1$ and $R' \equiv \mathcal{T}(Q', \Psi)$ applying (28). Thus $Q \overset{s}{\underset{\Psi}{\longmapsto}} Q'$ from Lemma 3 and by induction we have $Q' \models\!\!\models M_1$. As a result $Q \models\!\!\models \langle\beta\rangle M_1$. For $Q \models\!\!\models \langle\beta\rangle M_1$ the proof is similar. □

Proof of Proposition 5.
($\Rightarrow$) Let $P \boxminus Q$. Then $(P, Q) \in B$ which a must-bisimulation for $\underset{\Psi}{\dashrightarrow}$. We will show that $\mathfrak{T}(B, \Psi) =_{def} \{(P, \mathcal{T}(Q, \Psi)) \mid (P, Q) \in B\}$ is a bisimulation. If $P \xrightarrow{\alpha} P'$ then $Q \overset{s}{\underset{\Psi}{\longmapsto}} Q'$ where $\widehat{s} = \widehat{\alpha}$ and $(P', Q') \in B$ for some Q'. But then $(P', \mathcal{T}(Q', \Psi)) \in \mathfrak{T}(B, \Psi)$ and $\mathcal{T}(Q, \Psi) \xrightarrow{s} \mathcal{T}(Q', \Psi)$ what follows from Lemma 3. If $\mathcal{T}(Q, \Psi) \xrightarrow{\alpha} R'$ then we get $R' \equiv \mathcal{T}(Q', \Psi)$ (28) and applying Lemma 3 we have $Q \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q'$. Thus there exists P' such that $P \xrightarrow{s} P'$, $\widehat{s} = \widehat{\alpha}$ and $(P', Q') \in B$. Because then $(P', \mathcal{T}(Q', \Psi)) \in \mathfrak{T}(B, \Psi)$, $\mathfrak{T}(B, \Psi)$ is a bisimulation.
($\Leftarrow$) If $P \approx \mathcal{T}(Q, \Psi)$ then $(P, \mathcal{T}(Q, \Psi)) \in B$ which is the least bisimulation with this property. Applying (28), it is easy to see that there exists $C \subseteq \mathcal{P} \times \mathcal{P}_\Psi$ such that $B = \mathfrak{T}(C, \Psi)$ and it is enough to prove that C is a must-bisimulation. Let $(P, Q) \in C$ and $P \xrightarrow{\alpha} P'$. Then $(P, \mathcal{T}(Q, \Psi)) \in B$ and applying (28) there exists Q' and s such that $\mathcal{T}(Q, \Psi) \xrightarrow{s} \mathcal{T}(Q', \Psi)$, $\widehat{s} = \widehat{\alpha}$ and $(P', \mathcal{T}(Q', \Psi)) \in B$. Then we have $(P', Q') \in C$ and $Q \overset{s}{\underset{\Psi}{\longmapsto}} Q'$ from Lemma 3. Let $Q \overset{\alpha}{\underset{\Psi}{\longmapsto}} Q'$. Then $\mathcal{T}(Q, \Psi) \xrightarrow{\alpha} \mathcal{T}(Q', \Psi)$ from Lemma 3 and there exists P' and s such that $P \xrightarrow{s} P'$, $\widehat{s} = \widehat{\alpha}$ and $(P', \mathcal{T}(Q', \Psi)) \in B$. Thus $(P', Q') \in C$ what completes the proof that C is a must-bisimulation. □

C Proofs from Section 7.2

Proof of Lemma 6:
We proceed by induction on the length of s. Let $\Gamma_n =_{def} \Delta_n \oplus \Psi_n$.
($\Rightarrow$) For $s = \varepsilon$ it is immediate. Let $\langle X, \Delta\rangle \overset{\alpha:s}{\underset{\Psi}{\longmapsto}}{}^{i}_{j} \langle Y, \Delta\rangle$. Because the right-side expressions of Δ and Ψ are linear, applying (31) there exists Z and k such that $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle Z, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^{k}_{j} \langle Y, \Delta\rangle$. There are two cases:

- $\langle X, \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle$ where $k = 0$. Because Ψ has the proper effect on Δ, this transition can be only inferred from $\langle [\![\Delta]\!](X), \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle$. Thus we have $\langle [\![\Gamma_n]\!](X_i), \Gamma_n\rangle \xrightarrow{\alpha} \langle Z_0, \Gamma_n\rangle$ and by induction:
$$\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha} \widetilde{T}_0(\langle Z, \Delta\rangle, \Psi, n) \xrightarrow{s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$$
- $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\dashrightarrow}} \langle Z, \Delta\rangle$ where $i \neq n$ and $k = i+1$. It can be only inferred from $\langle [\![\Psi]\!](X), \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}} \langle Z, \Delta\rangle$ and so $\langle [\![\Psi]\!](X), \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle$ (Ψ is linear). Thus we have $\langle [\![\Gamma_n]\!](X_i), \Gamma_n\rangle \xrightarrow{\alpha} \langle Z_{i+1}, \Gamma_n\rangle$ because $i \neq n$ and by induction
$$\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha} \widetilde{T}_{i+1}(\langle Z, \Delta\rangle, \Psi, n) \xrightarrow{s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$$

($\Leftarrow$) For $s = \varepsilon$ it is immediate. Let $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha:s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$. Applying (31), $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha} \widetilde{T}_k(\langle Z, \Delta\rangle, \Psi, n) \xrightarrow{s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ for some Z and k. The first transition can be only inferred from $\langle [\![\Gamma_n]\!](X_i), \Gamma_n\rangle \xrightarrow{\alpha} \langle Z_k, \Gamma_n\rangle$ and there are two possible cases:

- $\langle [\![\Delta_n]\!](X_i), \Gamma_n\rangle \xrightarrow{\alpha} \langle Z_k, \Gamma_n\rangle$ where $k = 0$. Then $\langle [\![\Delta]\!](X), \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle$ and by induction we have $\langle X, \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^{0}_{j} \langle Y, \Delta\rangle$ and $\langle X, \Delta\rangle \overset{\alpha:s}{\underset{\Psi}{\longmapsto}}{}^{i}_{j} \langle Y, \Delta\rangle$.
- $\langle [\![\Psi_n]\!](X_i), \Gamma_n\rangle \xrightarrow{\alpha} \langle Z_k, \Gamma_n\rangle$ where $i \neq n$ and $k = i+1$. Then we have $\langle [\![\Psi]\!](X), \Delta\rangle \xrightarrow{\alpha} \langle Z, \Delta\rangle$ so $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\dashrightarrow}} \langle Z, \Delta\rangle$ (Ψ has the proper effect on Δ). Thus by induction, $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\dashrightarrow}} \langle Z, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^{i+1}_{j} \langle Y, \Delta\rangle$ so $\langle X, \Delta\rangle \overset{\alpha:s}{\underset{\Psi}{\longmapsto}}{}^{i}_{j} \langle Y, \Delta\rangle$. $\square$

Proof of Proposition 7:
It is enough to show that for all $i \in [0, n]$ and $M \in \mathcal{M}$:
$$\langle X, \Delta\rangle \models_i M \text{ iff } \widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \models M$$
The proof proceeds by induction on the structure of M.
($\Rightarrow$) Let $\langle X, \Delta\rangle \models_i \langle\beta\rangle M_1$. Then $\langle X, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^{i}_{j} Q'$ where $\widehat{s} = \beta$, $Q' \models_j M_1$ and by (31), $Q' \equiv \langle Y, \Delta\rangle$ for some $Y \in \mathcal{X}$. Thus $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ applying Lemma 6 and by induction, $\widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n) \models M_1$. Then because $\widehat{s} = \beta$, we finally have $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \models \langle\beta\rangle M_1$. In the remaining cases (*true*, $\neg M_1$ and $M_1 \wedge M_2$) the proof is obvious.
($\Leftarrow$) Let $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \models \langle\beta\rangle M_1$. Then $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{s} R'$ where $\widehat{s} = \beta$, $R' \models M_1$ and by (31), $R' \equiv \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ for some $Y \in \mathcal{X}$ and $j \in [0, n]$. Thus we have $\langle X, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^{i}_{j} \langle Y, \Delta\rangle$ applying Lemma 6, and by induction $\langle Y, \Delta\rangle \models_j M_1$. Finally we get $\langle X, \Delta\rangle \models_i \langle\beta\rangle M_1$ ($\widehat{s} = \beta$). The remaining cases are simple. $\square$

Proof of Proposition 8:
($\Rightarrow$) Let $P \mathrel{\sqsubset\!\!\!\!{\scriptstyle !}\;}_n \langle X, \Delta\rangle$. Then $(P, \langle X, \Delta\rangle) \in B_0$ where $\{B_i\}_{i=0}^n$ is the smallest conditional must-bisimulation with this property, and it is enough to show that

$$\widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n) \;=_{def}\; \bigcup_{i=0}^n \{(P, \widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n)) \mid (P, \langle X, \Delta\rangle) \in B_i\}$$

is a bisimulation. Let $i \in [0, n]$ and $(P, \widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n)) \in \widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n)$ where $(P, \langle X, \Delta\rangle) \in B_i$. We have:

- If $P \xrightarrow{\alpha} P'$ then $\langle X, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^i_j Q'$ where $\widehat{s} = \widehat{\alpha}$, $j \in [0, n]$, $(P', Q') \in B_j$ and $Q' \equiv \langle Y, \Delta\rangle$ applying (31). Thus $(P', \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)) \in \widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n)$ and $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{s} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ from Lemma 6.
- If $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha} R'$ then $R' \equiv \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ for some Y and j (31) and applying Lemma 6 we have $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}}{}^i_j \langle Y, \Delta\rangle$. Because $(P, \langle X, \Delta\rangle) \in B_i$, there exists P' and s ($\widehat{s} = \widehat{\alpha}$) such that $P \xrightarrow{s} P'$, $(P', \langle Y, \Delta\rangle) \in B_j$ and finally $(P', \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)) \in \widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n)$.

($\Leftarrow$) Let $P \approx \widetilde{T}(\langle X, \Delta\rangle, \Psi, n)$ and $(P, \widetilde{T}(\langle X, \Delta\rangle, \Psi, n)) \in B$ where B is the smallest bisimulation with this property. Applying (31), it is easy to see that there exists $\{B_i\}_{i=0}^n$ such that $B = \widetilde{\mathfrak{T}}(\{B_i\}_{i=0}^n, \Psi, n)$, and it is enough to show that $\{B_i\}_{i=0}^n$ is a conditional must-bisimulation. Let $(P, \langle X, \Delta\rangle) \in B_i$ where $i \in [0, n]$. Then $(P, \widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n)) \in B$ and we have:

- If $P \xrightarrow{\alpha} P'$ then $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{s} R'$ where $(P', R') \in B$, $\widehat{s} = \widehat{\alpha}$ and $R' \equiv \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$ for some Y and j (31). Thus $(P', \langle Y, \Delta\rangle) \in B_j$ and applying Lemma 6 we have $\langle X, \Delta\rangle \overset{s}{\underset{\Psi}{\longmapsto}}{}^i_j \langle Y, \Delta\rangle$.
- If $\langle X, \Delta\rangle \overset{\alpha}{\underset{\Psi}{\longmapsto}}{}^i_j Q'$ then $Q' \equiv \langle Y, \Delta\rangle$ (31) and applying Lemma 6 we have $\widetilde{T}_i(\langle X, \Delta\rangle, \Psi, n) \xrightarrow{\alpha} \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)$. Thus there exists P' and s ($\widehat{s} = \widehat{\alpha}$) such that $P \xrightarrow{s} P'$ and $(P', \widetilde{T}_j(\langle Y, \Delta\rangle, \Psi, n)) \in B$. Then it is enough to note that $(P', \langle Y, \Delta\rangle) \in B_j$. $\square$

This article was processed using the LaTeX macro package with LLNCS style