The Linear Time – Branching Time Spectrum II

The semantics of sequential systems with silent moves

Preliminary version

R.J. van Glabbeek[*]
Computer Science Department, Stanford University
Stanford, CA 94305, USA.
rvg@cs.stanford.edu

This paper studies semantic equivalences and preorders for sequential systems with silent moves, restricting attention to the ones that abstract from successful termination, stochastic and real-time aspects of the investigated systems, and the structure of the visible actions systems can perform. It provides a parameterized definition of a such a preorder, such that most such preorders and equivalences found in the literature are obtained by a suitable instantiation of the parameters. Other instantiations yield preorders that combine properties from various semantics. Moreover, the approach shows several ways in which preorders that were originally only considered for systems without silent moves, most notably the *ready simulation*, can be generalized to an abstract setting. All preorders come with—or rather as—a modal characterization, and when possible also a relational characterization. Moreover they are motivated by means of an (also parameterized) testing scenario, phrased in terms of 'button pushing experiments' on generative and reactive machines. The testing scenarios for *branching bisimulation*, *η-bisimulation* and *coupled simulation* and the corresponding modal characterizations are especially new.

Introduction

A general introduction to this line of research can be found in Van Glabbeek [17]. There a similar programme was carried out for finitely branching sequential systems without internal actions. Here this work is generalized to a setting of infinitely branching processes with silent moves, divergence and underspecification.

I will start with the testing scenarios in Section 1, leading to a parameterized notion of observability in Section 2. Section 3 contains the definition of the observable behaviour, according this notion, of systems represented as elements of a labelled transition space. This is where the technical part starts. Some readers may wish to read Sections 2 and 3 in parallel. For each notion of observability a *may* and a *must* preorder is defined in Section 4, such that the preceding formalization of observable behaviour constitutes its modal characterization. Then the preorders and their associated equivalences are partially ordered by inclusion through an exhaustive series of inclusion results and counterexamples thereof. In Section 5 relational characterizations of the preorders are provided when possible, and in Section 6 the preorders and equivalences found in the literature are positioned in the framework obtained.

1 Testing scenarios

In this section a variety of testing scenarios will be proposed, each determining a notion of observable behaviour of systems. In these testing scenarios the internal structure of systems is not considered observable; only the interactions of the system with the environment (observer) are. In order to visualize these interactions one can imagine something like the generative or reactive machine of Figure 1. The machine acts as a black box in which the investigated system can be placed. It has switches, displays and other gadgets that specify the interactions with the experimentator. How the machine reacts to stimuli received on these channels depends on the system inside, but the observer sees only the reactions of the machine without knowing how they are caused.

[*] This work was supported by ONR under grant number N00014-92-J-1974.

The machines employed in this paper are parameterized by a set A of observable actions, and are suitable only for the analysis of systems whose relevant behaviour can be expressed in terms of the actions $a \in A$ they can perform. Which activities of an investigated system are listed in A is a matter of choice, and depends on the level of abstraction at which one wants to analyze systems. Moreover, different activities that are indistinguishable on the chosen level of abstraction are interpreted as occurrences of the same action $a \in A$. Activities of the investigated system that are not listed in A are called *internal*. As internal actions are not distinguished, all of them will be denoted by the symbol $\tau \notin A$. In this paper the choice of A is fixed, suggesting (until Section 7) that only one level of abstraction will be considered. The reader should therefore keep in mind that all forthcoming testing scenarios, notions of observability, preorders and equivalences are in fact parameterized by this choice.

Throughout this paper I will restrict attention to the treatment of *uniform concurrency* [6]. This means that the internal structure of the actions $a \in A$ is not investigated. So it remains unspecified if these actions are in fact assignments to variables or the moves of a chess player or anything else.

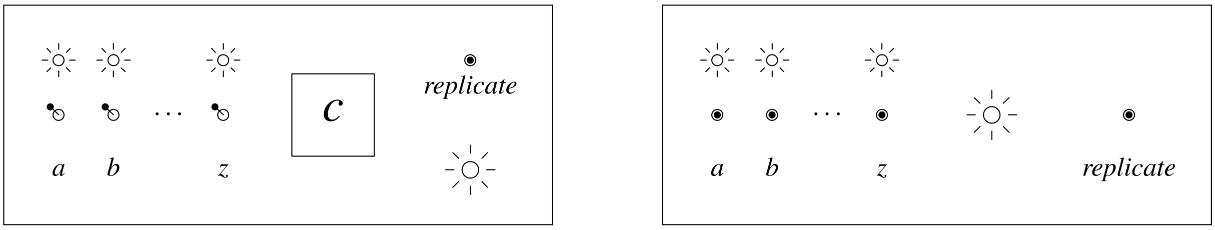

Figure 1: A generative and a reactive machine

In this section I distinguish *generative* and *reactive* systems. A generative system spontaneously performs actions, even if no input from the outside world is received. The only way to influence such a system is to *restrict* its possible behaviour. A reactive system on the other hand only acts in response to input from the environment. Whether an actual system should be classified as generative or as reactive is largely a matter of perspective. When a system needs to communicate (synchronously) with the environment, one can either take a successful communication for granted and consider an environment that does not permit the communication to be ‘restrictive’, or one could take the absence of communication to be the default and consider a successful communication to be a reaction on the act of the environment of permitting such a communication.

The *generative machine* of Figure 1 is thought to be an interface between generative systems and their environment. It has five features that specify its potential interactions with *the observer*.

- Its most basic ingredient is a *display*. Every time a visible action is generated by the system that is supposed to be lodging inside, the name of this action appears in the display. It stays there until the display is cleared for presenting another action. If the same action is generated twice in a row, clearing the display in between is considered observable. This allows the observer to record (a sequence of) visible actions performed by the investigated system.

- Secondly there is a *green light*, that allows one to record internal activity as well. It is supposed to be on when (internal) activity takes place, and off otherwise. In the latter case the investigated system is said to be *idle*.

- Thirdly there are several *menu lights*—one for every action $a \in A$. These show which visible actions could currently be performed.

- Furthermore the machine contains a series of switches—again one for every visible action. The observer can put each of these switches in one out of two positions: *free* or *blocked*. If a switch is put on *blocked*, the corresponding action may not be generated by the investigated system. This is the way in which, in the generative testing scenario, the behaviour of a system is restricted by its environment.

- Finally our machine has a *replication button*. Pressing this button results in a multitude of copies of the machine together with the residing system. In each of these copies the system will be in the same state as the original. The observer (and her friends) can now continue the session with each of these copies (in parallel).

Note that the lights and the display can be influenced by the system, but not directly by the environment, whereas the switches and the button can be influenced by the observer, but not by the investigated system. Thus input and output channels are separated.

The *reactive machine* serves as interface for reactive systems. Its lights and replication button play the same rôle as the ones of the generative machine. However, the display and the switches have been replaced by a series of *buttons*—one for every $a \in A$. As long as the buttons are not touched, the system rests. If the experimentator applies continuous pressure on the a-button, the system can react either by performing an a-action, in which case the button goes down, or by performing internal activity, possibly followed by an a-action. As long as internal activity goes on (maybe forever), the green light is on. After the a-button goes down, the system is in rest again and the light goes off, until either another button is pressed upon, or the a-button is released and put under pressure again. If the light goes off without the a-button going down, or if it doesn't go on in the first place, the a-experiment has 'failed' and the experimentator knows that the system has reached a state in which no a-action and no further internal activity is possible.

The reactive testing scenario, including the green light and the replication facility, stems form Milner [27]. The menus and the generative scenario originate from Van Glabbeek [17]. The term *generative* comes from Van Glabbeek, Smolka, Steffen & Tofts [18].

If the reactive testing scenario is expanded so as to allow the application of pressure on several buttons at the same time, it strongly resembles the generative testing scenario. The buttons that are not down and on which pressure is applied correspond to the switches that are put on *free*. Moreover the a-button going down corresponds to the name a appearing in the display. A difference is that in the reactive testing scenario a button that went down needs to be released before it can be used again. This would correspond to the property of switches to automatically jump to *blocked* whenever the corresponding action appears in the display. Under the assumption that the observer can rearrange the switches as she sees fit at any time, this change is of little significance, expect that it enables *alert replication* as explained below. Furthermore the state of the reactive machine in which no button receives pressure does not correspond to the state of the generative machine in which all switches are set on *blocked*. Namely in the former case not even internal activity is allowed, whereas in the latter case only the visible actions are blocked. In order to mimic this reactive state in the generative scenario, the generative machine should be equipped with an *on/off*-switch. This would not increase its testing power. Reversely, the state of the generative machine in which all visible activity is disabled corresponds to a state of the reactive machine in which only a button is put under pressure of which we know that the corresponding action cannot occur anyway.

Thus the reactive testing scenario can be regarded as a restricted version of the generative scenario in which only one switch at a time can be set *free*. From this point of view there is no further need to distinguish generative and reactive systems.

Now a variety of testing scenarios can be obtained by considering restricted versions of the machines described above, and variations in the capability of the observer to use them fully. To be precise, in this paper I consider a *testing scenario* to be fully determined by the positioning of checkmarks in the questionnaire below, one in each column.

<u>Switches or Buttons?</u>
□ No switches or buttons
□ Switches for blocking
□ Blocking and deblocking

<u>Lights?</u>
□ No lights
□ Green light
□ Menu lights

<u>Replication?</u>
□ No replication
□ In stable states
□ Everywhere
□ η-replication
□ Continuous

<u>Global testing?</u>
□ No global testing
□ In stable states
□ Everywhere, sloppy
□ Everywhere, weak
□ Everywhere, alert

<u>Finiteness?</u>
□ even finite switching
□ finite width & length
□ finite length only
□ all infinite
□ divergence sensitive

The first question asks whether or not the machine comes with switches (or buttons if one prefers the reactive viewpoint). If there are switches, one can consider a machine in which the observer is free to arrange the switches as she sees fit at any time, or one whose switches start out in the *free* position and can be put on *blocked*, but not back. The second questions asks whether the green light is present and if so if also the menu lights are present. I will not consider machines with menu lights but without the green light. The display is always present. The third question deals with the replication button. If it is present, one can distinguish the cases that is can be used in every state of the investigated system, or that replication is only possible when the system is idle (and the green light off). In the generative testing scenario the system can be idle only in a state where no internal activity is possible. Such a state is called *stable*. In case the observer makes a copy of a system immediately before a visible action happens, one may (or may not) assume that this can be noticed by the observer (by the fact that no time elapses between pressing the replication button and observing the action). The observer is then in the possession of a copy, known to be in the state where the visible action originated from. If this is possible I speak of η-replication, in the spirit of BAETEN & VAN GLABBEEK [5]. The last answer on the third question is that the observer may hold his thumb on the replication button during an extended period of time. This will create an ordered series of copies with the property that from each state the system has passed through, at least one copy is available. This possibility of *continuous copying* was independently suggested to me by Peter Weijland and Frits Vaandrager.

The fourth question deals with *global testing*, the scenario investigated in MILNER [27]. Here it is assumed that by making sufficiently many copies, *all* possible further behaviours of the investigated system can be recorded on one of the copies made. To this end one can assume that the behaviour of a system is completely determined by the status of the switches (buttons), and certain ambient conditions (the *weather*), and that it is possible to expose a copy of the machine to each possible weather condition [27]. By combining the observations made on each copy of the machine, it is possible to tell, not only that certain behaviours occurred, but also that certain behaviours did *not* occur. As in the previous question, one can choose if global testing is possible in stable states only, nowhere, or everywhere. As global testing presupposes replication it seems inconsistent to check the first or second box for question 3, and the third one for question 4. Still I will allow such answers as a restricted usage of replication. In case global testing is possible everywhere, I consider three subcases. My default assumption is that internal activity can happen between every two actions/observations of the experimentator. Thus, in order to ascertain that in a particular state some behaviours are possible and some others are impossible, one needs to make a lot of copies at once, some of which are used to see that a number of behaviours actually happen, while the rest is exposed to all weather conditions in order to ascertain that some behaviours do not happen. Under the *sloppy* scenario this requires two hits of the replication button (the second one to spread a copy over all weather conditions), and internal activity can take place between these hits. Under the *weak* scenario, all copies can be made at once. Another possibility is that the observer has such good reflexes that it can hit the replication button before any internal activity can take place. In particular, the observer can make copies of the system in a state that is known to be immediately after the occurrence of a visible action. This I call the *alert* scenario.

The last question actually combines three questions on the finiteness of observations. First of all one can choose that every time the replication button is pressed only *one* copy is made (or finitely many, which is equivalent), so that one could speak of a *duplication button*, or that at least as many copies are made as the branching degree of the investigated systems. The first alternative is incompatible with global testing (unless one restricts attention to finitely branching systems, as in [27], in which case both alternatives coincide). Therefore I will no further consider this combination. Also I will not investigate the possibility of a replication facility of infinite cardinality, smaller than the branching degree of the investigated systems. Secondly, one may ask if every time only finitely many switches can be put on free, or arbitrarily many. The first alternative, in combination with answer 3 on question 1, is the reactive scenario. Thirdly, one may ask if observations can last infinitely long. The option finitely long in combination with infinite replication only guarantees that each copy of the machine that participates in the experiment will be abandoned by its observer after a finite time. It does not guarantee that there is a global time at which the entire experiment is ended. Now if infinite replication is selected, I will assume that infinitely many switches can be set free as well. Moreover, I will not consider the combination of finite replication and infinite duration. This would give rise to countable replication as well, which for countably branching systems is as good as arbitrary replication, and for uncountably branching systems falls outside the scope of this study. This yields only four, linearly ordered answers on the three finiteness questions. In case observations may last forever, one could observe that from a certain point onwards nothing observable happened. In that case there is a choice of only recording what happened up to that point, which yields an observation that could have been made in a finite amount of time as well, or to explicitly add to the observation that nothing happened since. I call the latter option *divergence sensitive*, although it makes an infinite progression of internal activity (*divergence*) observable only in combination with the green light.

2 Notions of observability

In this section I describe the type of observations that can be made according to each of the testing scenarios of the previous section.

T In each of the testing scenarios it is possible to terminate the session at the machine at will. The observation that only consist of terminating the session is denoted T.

$a, a\epsilon$ In each of the testing scenarios it is possible to observe the occurrence of a visible action (by its appearance in the display, or by the corresponding button going down). The observation of the action $a \in A$, followed by the observation φ, is denoted $a\varphi$. My default assumption is that internal activity can happen before and after any observation. In some scenarios this assumption is partly given up, and one can distinguish between $a\varphi$ and $a\epsilon\varphi$, the latter explicitly allowing internal activity to happen between a and φ.

In the simplest testing scenario (the first box checked in every column), all observations are finite and of the form $a\varphi$ or T. Such an observation is a just a finite sequence of actions, ended by T. If observed during a session with system P, this sequence is called a *(partial) trace* of P, and the observable behaviour of P is given by the set of all its traces (that could have been observed). It need not to be assumed here that the machine can be restarted to observe several traces.

ϵ The observation $\epsilon\varphi$ says that φ can be observed after some (or none) internal activity. It is present in all testing scenarios. However, under the default assumption that internal activity can take place *anywhere*, $\epsilon\varphi$ is equivalent with φ and the operator ϵ is redundant.

0 If the green light is present, it is possible to observe that it is off, denoted by 0. This means that no activity goes on in the system. If moreover no switches or buttons are present to stimulate the system, it is difficult to image what on earth could resume activity in it. Hence I will assume that once the green light is off, it will stay that way and no further observations can be made. This adds a constant 0 to the language of observations.

With the constructions T, a and 0, one can built observations that look like a sequence of actions, ended by either T or 0. If a sequence of actions ending in 0 is observed, one knows that the investigated system (P) can perform the listed actions in succession and reach a state where no further activity is possible. Such a sequence is called a *complete(d) trace* of P.

FT If both the green light and the switches are present, and the switches can be manipulated freely, it *is* possible to recover from an idle period, and one can at least observe a sequence of actions interspersed with idle periods, denoted as a string over $A \cup 0$. Moreover, the observation can be enriched with information on the status of the switches at various points. Now I assume that internal activity happens independent of the positions of the switches. Thus the only relevant information about the switches is their position when an action appears in the display, or during an idle period. Suppose one observes for instance that first an action

a happens (the light must than be on), then b (the light is still on), then an idle period (the light is off) and then a again (the light must be on again). This observation can be denoted as $ab0aT$. Now if X_1, X_2 and X_5 are the sets of switches that were put *free* when a, b and a happened respectively, and X_3, X_4 are the sets of free switches during the first and second part of the idle period, the full observation can be denoted $a(X_1)b(X_2)0(X_3)0(X_4)c(X_5)T$. Now I make another assumption about the nature of the systems under investigation, namely that whenever $a(X)\varphi$ can be observed, a must be in X (that part is obvious) and for any other set Y of actions containing a, $a(Y)\varphi$ could have been observed as well. In other words, whether a can happen or not does not depend on the position of any other switch than the a-switch. As pointed out to me by Jan Bergstra, this rules out built-in priority mechanisms that for instance allow a to happen only if b cannot happen. Thus an observation $a(X)$ can without loss be shortened to a, and $0(X)$ will be abbreviated by $\tilde{X}$. If $\tilde{X}$ is observed one knows that the investigated system is unable to continue with either an internal action, or an action from X. The set X is then said to be *refused*, and called a *refusal set*. This adds a unary operator $\tilde{X}$ to the language of observations, for any $X \subseteq A$.

With these operators and the constructions a and T one can built observations that look like sequences over $A \cup \mathcal{P}(A)$. If observed during a session with system P, such a sequence is called a *failure trace* of P. In *failure trace semantics* (checkmarks 3-2-1-1-2), the observable behaviour of a system is given by its set of failure traces.

F In case the switches can not be reset from blocked to free, it is, as in the completed trace scenario, not possible to recover from an idle period. In combination with the green light an observation $\tilde{X}$ can be made, but in that case the machine reaches a state of deadlock. Thus the unary operation $\tilde{X}$ needs to be replaced by a constant $\tilde{X}$.

With these constants and the constructions a and T one can build observations that look like sequences ending in T or $\tilde{X}$. A sequence $\sigma\tilde{X}$ with $\sigma \in A^*$ and $X \subseteq A$, also denoted $<\sigma, X>$, is called a *failure pair*. In *failure semantics* (checkmarks 2-2-1-1-2), the observable behaviour of a system is given by the set of its failure pairs, its *failure set*, and the set of its traces.

R, RT If the menu lights are present I assume that they light up only during idle periods. Thus the observation 0 can be replaced by an observation $(X, \tilde{Y})$, where X is a set of actions whose light is seen to be burning, and $\tilde{Y}$ is a set of actions whose light is seen to be off. Of course the most informative observations are the ones where X and Y are complements, and by restricting attention to such observations, half of the pair $(X, \tilde{Y})$ is redundant. However, the presentation used here allows for smoother proofs later on, and generalizes easier to a finite setting. If the switches can be used for blocking only, there can be only one idle period in an observation, namely at the end, and a constant $(X, \tilde{Y})$, for $X, Y \subseteq A$, is added to the language of observations. If the switches are used for blocking as well as deblocking, one needs a unary operator $(X, \tilde{Y})$, that for convenience can be split in operators X and $\tilde{Y}$. In combination with these observations the information obtained from the position of the switches is redundant.

If there are no switches, an idle period can occur only when the menu of further actions is empty, so the presence of menu lights adds nothing to the observational powers of the machine. In the absence of the menu lights and the green light no refusals can be observed, and one is left with the observations T and a, independent of the answer on question one.

F^-, FT^-, R^-, RT^- The observations above exhaust all answers on the first two questions in case it is possible to set infinitely many switches free (answer 2–5 on question 5). In case only finitely many switches can be on *free*, one can make observations as in F or FT with X required to be finite. In combination with the menu lights, choosing answer 1 on question 5 can be interpreted as dealing with a machine in which the menu lights are also buttons that only light up if they are depressed (and the corresponding action is on the menu). By requiring that only one (or finitely many) lamp-buttons can be depressed at a time, one obtains observations as in R or RT, but with X and Y finite.

S The observation $S\varphi$ can be made if the system is in a stable state where it admits the observation φ. So far none of the testing scenarios discussed gave rise to this observation, but it will be handy later on. In order to cast this observation in the testing framework, one could add a question to the questionnaire from the previous section regarding the presence of a switch that simultaneously blocks all visible actions. If the green light goes off when this switch is engaged, a stable state has been reached. In the presence of switches for every action separately that can be manipulated in all directions, this universal switch is redundant, and the observation $S\varphi$ corresponds with $\widetilde{\emptyset}\varphi$. There is no testing scenario justifying observation S in the absence of the green light. In fact, the observation 0 (or $\widetilde{X}$) is made whenever S is observed and no visible action follows immediately (and X is the set of free switches).

$\wedge, \wedge\!\!\!\!{\scriptscriptstyle s}$ If a duplication button is present, one can make an observation φ on the original machine and an observation ψ on the duplicate, and combine these observations as $\varphi \wedge \psi$. This adds a binary operator $\wedge$ to the language in case one of the answers 3–5 is selected on question 3 and answer 1 or 2 on question 5. In case duplication is possible in stable states only, from the fact that duplication succeeds one can conclude that the machine is in a stable state, and an observation $S(\varphi \wedge \psi)$, shortened to $\varphi \wedge\!\!\!\!{\scriptscriptstyle s}\, \psi$, is obtained.

$\bigwedge, \bigwedge\!\!\!\!\!{\scriptscriptstyle s}$ In case infinite replication is selected (answer 2–5 on question 3 and 3–5 on 5), one obtains observations $\bigwedge_{i\in I}\varphi_i$ or $\bigwedge\!\!\!\!\!{\scriptscriptstyle s}_{\;i\in I}\varphi_i$ from the observations φ_i made on the respective copies. The size of the index sets I may be bounded by the branching degree of the systems considered.

η In case η-replication is selected one may do an observation $\varphi a \psi$. This differs from an observation $\varphi \wedge a\psi$, as the latter one allows internal activity to happen between the making the copy where φ will be observed and the occurrence of a. If $a\epsilon$ is selected instead of a, η-replication yields an observation $\varphi a\epsilon\psi$ instead of $\varphi a \psi$.

b In case continuous copying is selected it is possible to observe ψ after some period in which only internal activity takes place, and to observe φ in each copy of the system in a continuous period prior to ψ. This gives rise to an observation φ *just before* ψ, denoted $\varphi\tau\psi$. Continuous copying also enables η-replication of type $\varphi a\psi$. In Section 6 I will show that the observations η and b capture everything that can be observed with continuous copying.

∞ If observations may last forever one may observe infinite sequences of actions. In case switches or menus are available, these infinite sequences may be interspersed with observations $\widetilde{X}$, X or S. In combination with infinite replication one may propose that more complicated observations are possible, involving an infinity of nested conjunctions. It turns out however that such observations add nothing that is not observable without them. Therefore I will not consider them from the onset.

Δ, λ In the divergence sensitive scenarios there is an observation stating that nothing is observed for an infinite amount of time. In particular, it is not observed that the green light is off. If there is a green light, one can conclude that an infinite sequence of internal activity takes place (*divergence*), and the observation is denoted Δ. If there is no green light, one concludes that the system either diverges or reaches a state where no further activity is possible (*deadlock*). This observation is denoted λ.

$\overline{\neg}\!\!\!\!{\scriptscriptstyle s}\,, \overline{\neg}, \neg$ If global testing is selected and observations may last infinitely long, it is possible to do an observation $\neg\varphi$ whenever φ cannot be observed. Namely, expose the system to all possible weather conditions and make in each of the copies an observation that is different from φ. The negation operator comes in three flavours ($\overline{\neg}\!\!\!\!{\scriptscriptstyle s}\,$, $\overline{\neg}$, $\neg$), depending on whether global testing is possible only in stable states or everywhere, and in the latter case on whether or not the sloppy scenario is selected. In fact the definition of the operators $\overline{\neg}$ and $\neg$ will look the same, but in scenarios with $\neg$ the global assumption that internal activity can take place before *any* observation is given up. The difference between the *weak* and the *alert* replication scenarios (answers 4 and 5 on question 4) is made by choosing either a or $a\epsilon$ as action observations (see above). Under the sloppy scenario there is no difference between $a\varphi$ and $a\epsilon\varphi$.

$\bigvee$ The observation $\bigvee_{i\in I}\varphi_i$ can be made if one of the observations φ_i is made. It is merely an act of abstraction from the side of the observer and is redundant in all testing scenarios, except for sloppy global testing. Namely $\text{⫬}\bigvee_{i\in I}\varphi_i$ is a stronger observation then $\bigwedge_{i\in I}\text{⫬}\varphi_i$. The former observation says that after some internal activity one may reach a state where one can observe that none of the observations φ_i can be made, whereas the latter says that for every observation φ_i it is possible to perform some internal activity and reach a state where one can observe that φ_i can not be observed. I prefer the stronger interpretation of the sloppy global testing scenario, and $\bigvee$ will for convenience be part of all testing scenarios.

Each of the items listed above contributes to the language of all possible observations. A notion of observability is completely determined by the choice of these items and a selection of closure properties, and this choice is completely determined by the chosen testing scenario. Here I consider the following closure properties:

τ This closure property says that internal activity can happen before any observation that does not start with $\neg$, $\wedge$, $\bigwedge$ or $\bigvee$. The set of all observations that can be preceded by internal activity, i.e. are not of the form $\neg\varphi$ etc., is denoted $\mathbb{O}(\tau)$. All notions of observability of this paper have this closure property.

$\not\lambda, \not\Delta$ If global testing is selected but observations are of finite length, it is not always possible to observe $\neg\varphi$ if φ cannot be observed. Namely if the system can diverge, by never performing an observable action nor putting off the green light, the diverging copy of the replicated system will never reach a state in which one knows that it will produce an observation different from φ. Now the closure property $\not\Delta$ inhibits the observation $\neg\varphi$ (resp. $\text{⫬}\varphi$ or $\neg_s\varphi$) for systems that can diverge, at least if $\varphi\in\mathbb{O}(\tau)$. It should only be selected if divergence is not observable, but deadlock is. Similarly, closure property $\not\lambda$ inhibits such observations for systems that can diverge or deadlock. It should be selected only if neither deadlock nor divergence is observable.

Definition 1 A *notion of observability* is a subset of

$$\{T,\ a,\ \tau,\ a\epsilon,\ \epsilon,\ \bigvee,\ 0,\ F^{(-)},\ FT^{(-)},\ R^{(-)},\ RT^{(-)},\ S,\ \wedge,\ \wedge_s,\ \bigwedge,\ \bigwedge_s,\ \eta,\ b,\ \infty,\ \Delta,\ \lambda,\ \neg_s,\ \text{⫬},\ \neg,\ \not\Delta,\ \not\lambda\}$$

(1) always containing T, (a or $a\epsilon$), τ, ϵ and $\bigvee$, and one of λ, Δ, $\not\lambda$, $\not\Delta$, $\neg_s$, ⫬ and $\neg$.
(2) containing at most one entry from 0, F, FT, R, F^-, FT^-, R^-, the *completed observations*,
(3) not containing RT without FT, RT^- without FT^-, b without η and a, or η without $\bigwedge$ or $\wedge$,
(4) not containing one of ∞, $\neg$, ⫬, $\neg_s$, $\bigwedge$, $\bigwedge_s$ with one of $\wedge$, $\wedge_s$, F^-, FT^-, R^-, RT^-,
(5) not containing λ or Δ without ∞, or Δ (λ) with $\not\Delta$($\not\lambda$), or a completed observation with λ or $\not\lambda$,
(6) and not containing $\Delta, \not\Delta$ or a *stable observation* (S, $\wedge_s$, $\bigwedge_s$, $\neg_s$) without a completed observation.

Table 1 proposes a name and abbreviation thereof for each of these notions. Check which of the 20 listed combinations of observations occurs in N, and make a name (abbreviation) by combining the corresponding keywords (abbreviations). There may be several keywords from the first column, but at most one from the second, namely the first one that applies. Redundant parts, such as ∞ in the presence of Δ, may be omitted. It will follow from Theorem 1 that notions with the same name yield identical preorders.
For each notion of observability N, the class $\mathbb{O}_N$ of *potential N-observations* is defined inductively by the clauses $N-\{\tau,\not\Delta,\not\lambda\}$ listed in this section. If for example $N=\{T,a,\tau,\epsilon,\bigvee,\bigwedge,FT,\eta,\infty,\Delta\}$, i.e. divergence sensitive stability respecting failure η-simulation ($FS^{s\eta\Delta}$), then $\mathbb{O}_N$ is the smallest class satisfying

- $T,\Delta\in\mathbb{O}_N$ and $\sigma\in\mathbb{O}_N$ for σ an infinite sequence over $A\cup\{\widetilde{X}\mid X\subseteq A\}$;
- if $\varphi,\psi\in\mathbb{O}_N$, $a\in A$ and $X\subseteq A$, then $a\varphi\in\mathbb{O}_N$, $\widetilde{X}\varphi\in\mathbb{O}_N$, $\epsilon\varphi\in\mathbb{O}_N$ and $\varphi a\psi\in\mathbb{O}_N$;
- and if $\varphi_i\in\mathbb{O}_N$ for $i\in I$ then $\bigvee_{i\in I}\varphi_i\in\mathbb{O}_N$ and $\bigwedge_{i\in I}\varphi_i\in\mathbb{O}_N$.

F^-, FT^-, R^- or RT^-	finite	$^-$	b, η, a and $\neg$	branching bisimulation	*(BB)*
$\wedge$ or $\wedge_s$	finitary	*	$\neg$ and a	delay bisimulation	*(DB)*
∞	infinitary	$^\infty$	$\neg$ and $a\epsilon$	weak bisimulation	*(WB)*
Δ or λ	divergence sensitive	$^\Delta,^\lambda$	⫬ and $\bigwedge$	coupled simulation	*(CS)*
$\not\Delta$ (or $\not\lambda$)	(strongly) convergent	$^\downarrow,^{\downarrow\downarrow}$	⫬$_s$ and $\bigwedge$	stably coupled simulation	*(SCS)*
S, FT or RT	stability respecting	^{s}	⫬ but not $\bigwedge$	contrasimulation	*(C)*
η but not b	η-	$^\eta$	⫬$_s$ but not $\bigwedge$	stable bisimulation	*(SB)*
0	completed	^{0}	$\bigwedge$ or $\wedge$	simulation	*(S)*
F, F^-, FT or FT^-	failure	F	$\bigwedge_s$ or $\wedge_s$	stable simulation	*(SS)*
R, R^-, RT or RT^-	ready	R	not F or R	trace	*(T)*

Table 1: Naming convention

3 Observable behaviour

This section describes the observable behaviour of systems that are represented as elements of a labelled transition space, according to each of the notions of observability discussed before.

Definition 2 A *(κ-bounded) partial labelled transition space (PLTS)* is a triple $(\mathbb{P}, \rightarrow, \uparrow)$ with $\mathbb{P}$ a class (of *processes*), $\uparrow \subseteq \mathbb{P}$ (the *underspecified* processes) and $\rightarrow \subseteq \mathbb{P} \times Act \times \mathbb{P}$ for Act a set (of *actions*), such that for $p \in \mathbb{P}$ and $\alpha \in Act$ the class $\{q \in \mathbb{P} \mid p \xrightarrow{\alpha} q\}$ is a set (of cardinality $< \kappa$).

Notation: Write $p \xrightarrow{\alpha} q$ for $(p, \alpha, q) \in \rightarrow$ and $p \xrightarrow{\alpha}$ for $\exists q \in \mathbb{P} : p \xrightarrow{\alpha} q$. Not $p \uparrow$ is denoted $p \downarrow$.

I will consider transition spaces labelled over $Act = A \dot\cup \{\tau\}$. The elements of $\mathbb{P}$ represent the systems we are interested in, and $p \xrightarrow{\alpha} q$ means that system p can evolve into system q while performing the action α. I use the word 'space' instead of 'system' to emphasize that the elements of an PLTS are *all* systems under investigation, and not merely the states of a single system. It is common to assume that the transitions are instantaneous, but here it suffices to assume that they take a finite amount of time. Furthermore, in every state a finite amount of internal activity can take place (or none at all) without any transition being performed, but after this activity the system *has* to take one of its outgoing transitions, unless they are all labelled with actions that are blocked by the environment (or that have buttons which are not under pressure). This is known as a *progress assumption.* Systems that (also) have states in which an infinite amount of internal activity can take place, can be modelled by attaching a τ-loop ($p \xrightarrow{\tau} p$) to those states. Finally I assume that in a finite time only finitely many states are passed through, i.e. I consider *discrete* systems only.

My main interest is in *total* LTSs where $\uparrow = \emptyset$. Here the convention applies that *all* (outgoing) transitions of the represented systems actually appear in the LTS. However, sometimes only partial information on the represented systems is available, and then a partial LTS is used, that only contains the transitions of whose presence one is certain. If there is no certainty that all transitions leaving a given state are represented, this state is labelled with the symbol $\uparrow$.

For a total LTS $\mathbb{P}$ and $p \in \mathbb{P}$, one could now define the set $\mathcal{O}_N(p)$ of observations that can be made when p is placed in the machine that gives rise to the notion N of observability. But for a partial LTS there is insufficient information to do so, as there may be transitions that are not specified in $\mathbb{P}$. However, one can define a lowerbound and an upperbound of the set of observations that can be made, depending on the extra outgoing transitions of underspecified processes.

Definition 3 Let $(\mathbb{P}, \rightarrow, \uparrow)$ be a (κ-bounded) PLTS, labelled over $Act = A \dot\cup \{\tau\}$, and let N be a notion of observability. The functions $\mathcal{O}_N, \mathcal{H}_N : \mathbb{P} \rightarrow \mathcal{P}(\mathbb{O}_N)$ of *definite* and *hypothetical* N-*observations* of a process are inductively defined by the clauses below. The clauses for $\mathcal{O}_N$ are the ones above the line that are marked with an element of $N \cup \{\tau\}$, whereas the ones for $\mathcal{H}_N$ are the same clauses with the rôles of $\mathcal{O}$ and $\mathcal{H}$ exchanged, together with the ones under the line that are marked with an element of $N \cup \{\uparrow\}$. (Thus the clauses τ and $\uparrow$ are used regardless of N.)

(T)	T	$\in \mathcal{O}_N(p)$	
(a)	$a\varphi$	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{a} q$	$\wedge\ \varphi \in \mathcal{O}_N(q)$
(τ)	φ	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{\tau} q$	$\wedge\ \varphi \in \mathcal{O}_N(q) \wedge \varphi \in \mathbb{O}(\tau)$
(ϵ)	$\epsilon\varphi$	$\in \mathcal{O}_N(p)$ if	$\varphi \in \mathcal{O}_N(p)$
(0)	0	$\in \mathcal{O}_N(p)$ if $p \overset{\alpha}{\not\longrightarrow}$ for $\alpha \in Act \wedge p\downarrow$	
(F)	$\tilde{X}$	$\in \mathcal{O}_N(p)$ if $p \overset{\alpha}{\not\longrightarrow}$ for $\alpha \in X \cup \{\tau\} \wedge p\downarrow$	
(FT)	$\tilde{X}\varphi$	$\in \mathcal{O}_N(p)$ if $p \overset{\alpha}{\not\longrightarrow}$ for $\alpha \in X \cup \{\tau\} \wedge p\downarrow$	$\wedge\ \varphi \in \mathcal{O}_N(p)$
(R)	$(X, \tilde{Y})$	$\in \mathcal{O}_N(p)$ if $p \overset{\alpha}{\not\longrightarrow}$ for $\alpha \in Y \cup \{\tau\} \wedge p\downarrow$	$\wedge\ p \xrightarrow{a}$ for $a \in X$
(RT)	$X\varphi$	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{a}$ for $a \in X \wedge p \overset{\tau}{\not\longrightarrow} \wedge p\downarrow$	$\wedge\ \varphi \in \mathcal{O}_N(p)$
$(\wedge)$	$\varphi \wedge \psi$	$\in \mathcal{O}_N(p)$ if $\varphi, \psi \in \mathcal{O}_N(p)$	
$(\bigwedge)$	$\bigwedge_{i \in I} \varphi_i$	$\in \mathcal{O}_N(p)$ if $\varphi_i \in \mathcal{O}_N(p)$ for all $i \in I$	
$(\bigvee)$	$\bigvee_{i \in I} \varphi_i$	$\in \mathcal{O}_N(p)$ if $\varphi_i \in \mathcal{O}_N(p)$ for some $i \in I$	
(S)	$S\varphi$	$\in \mathcal{O}_N(p)$ if $p \overset{\tau}{\not\longrightarrow} \wedge p\downarrow$	$\wedge\ \varphi \in \mathcal{O}_N(p)$
$(\wedge_s)$	$\varphi \wedge_s \psi$	$\in \mathcal{O}_N(p)$ if $\varphi, \psi \in \mathcal{O}_N(p)$	$\wedge\ p \overset{\tau}{\not\longrightarrow} \wedge p\downarrow$
$(\bigwedge_s)$	$\bigwedge_{s\, i \in I} \varphi_i$	$\in \mathcal{O}_N(p)$ if $\varphi_i \in \mathcal{O}_N(p)$ for all $i \in I$	$\wedge\ p \overset{\tau}{\not\longrightarrow} \wedge p\downarrow$
(η)	$\varphi a \psi$	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{a} q \quad \wedge \quad \varphi \in \mathcal{O}_N(p)$	$\wedge\ \psi \in \mathcal{O}_N(q)$
(b)	$\varphi \tau \psi$	$\in \mathcal{O}_N(p)$ if $p = q$ or $p \xrightarrow{\tau} q \wedge \varphi \in \mathcal{O}_N(p)$	$\wedge\ \psi \in \mathcal{O}_N(q)$
(∞)	σ	$\in \mathcal{O}_N(p)$ if σ is an infinite sequence over $\{a, \tilde{X}, X, S\}$ and for any suffix ρ of σ, '$\sigma \in \mathcal{O}_N(p)$' is derivable from '$\rho \in \mathcal{O}_N(p)$' using the clauses in $N - \{\infty\}$.	
(Δ)	Δ	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots$	
(λ)	λ	$\in \mathcal{O}_N(p)$ if $p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots$	$\vee\ p \overset{\alpha}{\not\longrightarrow}$ for $\alpha \in Act$
$(\neg_s)$	$\neg_s\varphi$	$\in \mathcal{O}_N(p)$ if $\varphi \notin \mathcal{H}_N(p)$	$\wedge\ p \overset{\tau}{\not\longrightarrow} \wedge p\downarrow$
$(\bar{\neg})$	$\bar{\neg}\varphi$	$\in \mathcal{O}_N(p)$ if $\varphi \notin \mathcal{H}_N(p)$	
$(\neg)$	$\neg\varphi$	$\in \mathcal{O}_N(p)$ if $\varphi \notin \mathcal{H}_N(p)$	
$(\uparrow)$	φ	$\in \mathcal{H}_N(p)$ if $p\uparrow$	$\wedge\ \varphi \in \mathbb{O}(\tau)$
$(\not\Delta)$	φ	$\in \mathcal{H}_N(p)$ if $p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots$	$\wedge\ \varphi \in \mathbb{O}(\tau)$
$(\not\lambda)$	φ	$\in \mathcal{H}_N(p)$ if $(p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots \vee p \overset{a}{\not\longrightarrow}$ for $a \in Act) \wedge \varphi \in \mathbb{O}(\tau)$	

Here $a \in A$, $X, Y \subseteq A$ and I a set (of cardinality $< \kappa$).
There are also clauses F^-, FT^-, R^- and RT^-, defined as above, but with X and Y finite.
Futhermore the clauses for $a\epsilon\varphi$ and $\varphi a\epsilon\psi$ (if $a\epsilon \in N$) are obtained as special cases of a and η.

Most clauses in this definition should be sufficiently motivated by the previous section. Clause (τ) says that internal activity can happen before any observation that does not start with $\neg$, $\wedge$, $\bigwedge$ or $\bigvee$. If $\neg \notin N$, the condition $\varphi \in \mathbb{O}(\tau)$ can equivalently be omitted. As menu lights are assumed to be operational only in stable states, one has a condition $p \overset{\tau}{\not\longrightarrow}$ in the clauses (R), (RT), etc., just as in the clauses (0), (F), etc. In an underspecified state p one can not be sure that there is no outgoing τ-transition, so no definite observations (R), (F), etc. can be made, which explains the conditions $p\downarrow$ in those clauses. On the other hand, for *any* observation φ there could very well be a τ-transition to a state where φ can be observed. Therefore, clause $(\uparrow)$ introduces φ as a hypothetical observation of p, provided that it admits internal activity to preceed it. Clause $(\neg)$ expresses that $\neg\varphi$ is a definite observation iff φ is not even a hypotetical observation, and likewise that $\neg\varphi$ is a hypothetical observation iff φ is not a definite one.

Suppose that a process can be restarted from its initial state arbitrarily often, and can at those occasions be exposed to all possible whether conditions. In that case it is not only possible to find out that certain observations can be made, but also that certain observations cannot be made. However, if the closure property $\not\Delta$ or $\not\lambda$ is selected, it may be that an observation cannot be made, without this being observable. In this situation the class of hypothetical observations is extended to all those observations for which it is not possible to find out that they cannot be made. In case of divergence, it is for no observation $\varphi \in \mathbb{O}(\tau)$ possible to find out that it cannot be made, so all these observations belong to $\mathcal{H}_N(p)$.

4 May and must preorders and equivalences

Let N be a notion of observability. Two systems p can q are related by the *N-may preorder* if whenever p may pass a test, q may pass this test too; they are related by the *N-must preorder* if whenever p must pass a test, q must do so too. Here a test can be understood as a partular type of interaction with the observer, imposed by the latter by manipulating the appropriate machine, together with a designated set of observations resulting from this interaction, say the observations $\{\varphi_i \mid i \in I\}$, that qualify as *passing* the test. p *may pass* the test if one of the observations φ_i can be made, i.e. if $\bigvee_{i \in I} \varphi_i \in \mathcal{O}_N(p)$. p *must pass* the test, if (it can be observed that) no alternative to the desired outcome can occur. Suppose for instance that the test consist of waiting until a appears in the display, then blocking all actions except for b and c, and waiting until b happens, and is defined to succeed if either something else than a happens first, or b actually happens right after a, then p *must* pass this test iff p admits neither an observation acT nor $\widetilde{a\{b\}}$, *and* p cannot divergence in its initial state or right after a. Thus for every *may* test there is an observation $\varphi \in \mathbb{O}_N$ (possibly a disjunction) such that p passes the test iff $\varphi \in \mathcal{O}_N(p)$, and for every *must* test there is an observation $\psi \in \mathbb{O}_N(p)$ (possibly a disjunction) such that p passes the test iff $\psi \notin \mathcal{H}_N(p)$. In order to truly *observe* that p passes a *must* test, one has to assume a form of global testing that makes it possible to observe *all* runs of the investigated system from its initial state.

Definition 4 Two processes $p, q \in \mathbb{P}$ are *N-equivalent*, denoted $p =_N q$, if $\mathcal{O}_N(p) = \mathcal{O}_N(q)$.
p is *N-(may) prequivalent* to q, denoted $p \sqsubseteq_N^{\text{may}} q$, or simply $p \sqsubseteq_N q$, if $\mathcal{O}_N(p) \subseteq \mathcal{O}_N(q)$.
p is *N-must prequivalent* to q, denoted $p \sqsubseteq_N^{\text{must}} q$, if $\mathcal{H}_N(q) \subseteq \mathcal{H}_N(p)$.
p is *N-may-and-must prequivalent* to q, denoted $p \sqsubseteq_N^{\text{mm}} q$, if $p \sqsubseteq_N^{\text{may}} q$, and $p \sqsubseteq_N^{\text{must}} q$.
N-must equivalence and *N-may-and-must equivalence* are defined as expected.

Note that $\mathcal{O}_N(p) \subseteq \mathcal{H}_N(p)$ for any $p \in \mathbb{P}$, so $p \sqsubseteq_N^{\text{mm}} q$ iff $\mathcal{O}_N(p) \subseteq \mathcal{O}_N(q) \subseteq \mathcal{H}_N(q) \subseteq \mathcal{H}_N(p)$.

In case $\Delta, \lambda, \not\Delta, \not\lambda \notin N$, the relation $\sqsubseteq_N^{\text{must}}$ is not a very useful one, as, in terms of the above example, it is inpossible to require that p cannot diverge in its initial state or right after a. If moreover $\neg_s, \overline{\neg}, \neg \notin N$, the relation $\sqsubseteq_N^{\text{may}}$ coincides with $\sqsubseteq_{N \cup \{\Delta\}}^{\text{may}}$ or $\sqsubseteq_{N \cup \{\lambda\}}^{\text{may}}$. For this reason such N could be excluded from further investigation in Definition 1.

For N and M notions of observability write $N \preceq M$ if $p \sqsubseteq_M^{\text{may}} q \Rightarrow p \sqsubseteq_N^{\text{may}} q$ as well as $p \sqsubseteq_M^{\text{must}} q \Rightarrow p \sqsubseteq_N^{\text{must}} q$ for every LTS $\mathbb{P}$ and $p, q \in \mathbb{P}$. For S and T subsets of notions of observability write $S \preceq^c T$ if $N \preceq M$ for any notions N and M where M is N with S replaced by T.

Proposition 1 Let N and M be notions of observability with $N \cap \{\not\Delta, \not\lambda\} = M \cap \{\not\Delta, \not\lambda\}$. Then

$$N \subseteq M \text{ implies } N \preceq M.$$

Proof: Trivial, considering that $\mathcal{O}_N(p) = \mathcal{O}_M(p) \cap \mathbb{O}_N$ and $\mathcal{H}_N(p) = \mathcal{H}_M(p) \cap \mathbb{O}_N$. □

Proposition 2 $\{\not\lambda\} \preceq^c \{\not\Delta, \bigvee, 0\}$, $\{\not\lambda\} \preceq^c \{\lambda, \bigvee\}$ and $\{\not\Delta\} \preceq^c \{\Delta, \bigvee\}$.

Proof: It suffices to provide translations $f, g : \mathbb{O}_N \to \mathbb{O}_M$, such that $\varphi \in \mathcal{O}_N(p)$ iff $f(\varphi) \in \mathcal{O}_M(p)$ and $\varphi \in \mathcal{H}_N(p)$ iff $g(\varphi) \in \mathcal{H}_M(p)$. For the first statement f and g are recursively defined by

$$\begin{array}{rclcrclc} f(\neg\varphi) & = & \neg g(\varphi) & \quad & g(\neg\varphi) & = & \neg f(\varphi) & \quad \text{(and similarly for } \neg_s, \overline{\neg} \text{ and } \neg\text{)} \\ f(\Gamma\vec{\varphi}) & = & \Gamma f(\vec{\varphi}) & & g(\Gamma\vec{\varphi}) & = & 0 \vee \Gamma g(\vec{\varphi}) & \quad \text{(for any other observation } \Gamma\vec{\varphi}\text{)} \end{array}$$

With induction on φ one checks that this translation has the required property. Two typical steps, in which $\Longrightarrow$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$, are:

$$\neg\varphi \in \mathcal{O}_N(p) \;\Leftrightarrow\; \varphi \notin \mathcal{H}_N(p) \overset{\text{induction}}{\Leftrightarrow} g(\varphi) \notin \mathcal{H}_M(p) \;\Leftrightarrow\; \neg g(\varphi) \in \mathcal{O}_M(p)$$

$$a\varphi \in \mathcal{H}_N(p) \;\Leftrightarrow\; (p \Longrightarrow \xrightarrow{a} q \wedge \varphi \in \mathcal{H}_N(q)) \vee (p \Longrightarrow q \wedge q \stackrel{\alpha}{\not\longrightarrow} (\alpha \in Act)) \vee p \xrightarrow{\tau}\xrightarrow{\tau} \cdots \overset{\text{induction}}{\Leftrightarrow}$$

$$(p \Longrightarrow \xrightarrow{a} q \wedge g(\varphi) \in \mathcal{H}_M(q)) \vee (p \Longrightarrow q \wedge q \stackrel{\alpha}{\not\longrightarrow} (\alpha \in Act)) \vee p \xrightarrow{\tau}\xrightarrow{\tau} \cdots \;\Leftrightarrow\; 0 \vee ag(\varphi) \in \mathcal{O}_M(p)$$

For the other two statements 0 should be replaced by λ and Δ respectivily. □

Proposition 3 $\{0\} \preceq^c \{F\} \preceq^c \{FT, T\}$, $\{F\} \preceq^c \{R\} \preceq^c \{RT, FT, T\}$, $\{S\} \preceq^c \{FT\} \preceq^c \{F, \wedge^{s}\}$, $\{RT\} \preceq^c \{R, \wedge^{s}\}$, $\{S\} \preceq^c \{\wedge^{s}\} \preceq^c \{S, \wedge\}$, $\{\wedge^{s}\} \preceq^c \{\bigwedge^{s}\} \preceq^c \{S, \bigwedge\}$, $\{\wedge\} \preceq^c \{\bigwedge\}$, $\{\lambda\} \preceq^c \{\Delta, \bigvee, 0\}$, $\{RT\} \preceq^c \{\wedge^{s}, \bigwedge^{s}, a, T\}$, $\{F\} \preceq^c \{\bigwedge^{s}, \neg^{s}, a, T\}$, $\{\bigwedge^{s}\} \preceq^c \{\neg^{s}, \bigvee\}$, $\{S\} \preceq^c \{\neg^{s}\} \preceq^c \{S, ⫬\}$, $\{S\} \preceq^c \{⫬, \bigvee, F\}$, $\{S\} \preceq^c \{⫬, \bigvee, \Delta\}$, $\{\neg^{s}, \not\Delta\} \preceq^c \{⫬, \not\Delta\} \preceq^c \{\neg^{s}, \not\Delta\}$, $\{\bigwedge\} \preceq^c \{\neg, \bigvee\}$ & $\{⫬\} \preceq^c \{\epsilon, \neg\}$.

Proof: This follows from the following translations, that also hold with $\mathcal{O}$ replaced by $\mathcal{H}$.

- $0 \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \tilde{A} \in \mathcal{O}_M(p)$
- $\tilde{X} \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \tilde{X}T \in \mathcal{O}_M(p)$
- $\tilde{X} \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad (\tilde{X}, \emptyset) \in \mathcal{O}_M(p)$
- $(X, \tilde{Y}) \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad X\tilde{Y}T \in \mathcal{O}_M(p)$
- $S\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \tilde{\emptyset}\varphi \in \mathcal{O}_M(p)$
- $\tilde{X}\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \tilde{X}\wedge^{s}\varphi \in \mathcal{O}_M(p)$
- $X\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad (X, \tilde{\emptyset})\wedge^{s}\varphi \in \mathcal{O}_M(p)$
- $S\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \varphi\wedge^{s}\varphi \in \mathcal{O}_M(p)$
- $\varphi\wedge^{s}\psi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad S(\varphi \wedge \psi) \in \mathcal{O}_M(p)$
- $\varphi_1\wedge^{s}\varphi_2 \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \bigwedge^{s}_{i\in\{1,2\}}\varphi_i \in \mathcal{O}_M(p)$
- $\bigwedge^{s}_{i\in I}\varphi_i \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad S(\bigwedge_{i\in I}\varphi_i) \in \mathcal{O}_M(p)$
- $\varphi_1 \wedge \varphi_2 \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \bigwedge_{i\in\{1,2\}}\varphi_i \in \mathcal{O}_M(p)$
- $\lambda \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \Delta \vee 0 \in \mathcal{O}_M(p)$
- $X\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \varphi\wedge^{s} \bigwedge^{s}_{a\in X} aT \in \mathcal{O}_M(p)$
- $\tilde{X} \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \bigwedge^{s}_{a\in X}\neg^{s} aT \in \mathcal{O}_M(p)$
- $\bigwedge^{s}_{i\in I}\varphi_i \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \neg^{s}\bigvee_{i\in I}\neg^{s}\varphi_i \in \mathcal{O}_M(p)$
- $S\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \neg^{s}\neg^{s}\varphi \in \mathcal{O}_M(p)$
- $\neg^{s}\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad S⫬\varphi \in \mathcal{O}_M(p)$
- $S\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad ⫬(⫬\varphi \vee ⫬\tilde{\emptyset}) \in \mathcal{O}_M(p)$
- $S\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad ⫬(⫬\varphi \vee \Delta) \in \mathcal{O}_M(p)$
- $\neg^{s}\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad ⫬\varphi \in \mathcal{O}_M(p)\ (\not\Delta \in N, M)$
- $\bigwedge_{i\in I}\varphi_i \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \neg\bigvee_{i\in I}\neg\varphi_i \in \mathcal{O}_M(p)$
- $⫬\varphi \in \mathcal{O}_N(p) \quad \Leftrightarrow \quad \epsilon\neg\varphi \in \mathcal{O}_M(p)$

The appearence of ϵ in the last line is needed to assure that the translation preserves membership of $\mathbb{O}(\tau)$, so that the effect of clauses τ, $\uparrow$, $\not\Delta$ and $\not X$ on both formulae will be the same. □

Proposition 4 $\{F^-\} \preceq^c \{F\}$, $\{FT^-\} \preceq^c \{FT\}$, $\{R^-\} \preceq^c \{R\}$, $\{RT^-\} \preceq^c \{RT\}$ and the statements of Proposition 3 with F, FT, R and RT also hold with F^-, FT^-, R^- and RT^- respectivily. Moreover $\{RT^-\} \preceq^c \{\wedge^{s}, a, T\}$ and $\{\infty\} \preceq^c \{\bigwedge\}$.
Proof: Trivial or exactly as in the previous proof. The last one follows from Theorem 2. □

Proposition 5 $\{a\epsilon\} \preceq^c \{a, \epsilon\}$ and when $\neg \notin N$, then $\{\epsilon\} \cup N \preceq N$, $\{a\epsilon\} \cup N \preceq \{a\} \cup N$ and $\{b\} \cup N \preceq \{\wedge\} \cup N$.
Proof: If $\neg \notin N$, then $\epsilon\varphi \in \mathcal{O}_{N\cup\{\epsilon\}}(p) \Leftrightarrow \varphi \in \mathcal{O}_N(p)$, as follows with a straightforward induction on φ. Furthermore $(\epsilon\varphi)\tau(\epsilon\psi) \in \mathcal{O}_{N\cup\{b\}}(p) \Leftrightarrow \epsilon\varphi \wedge \epsilon\psi \in \mathcal{O}_N(p)$. □

The results of this section are visualized in Figure 2. Every notion of observability corresponds with a dot in this framework, that can be found with the naming convention of Table 1. The lines in the picture (dotted and dashed lines have no special meaning) can be regarded as arrows pointing upwards (unless there is an explicit arrowhead indicating otherwise), and $N \preceq M$ if there is a path from N to M, not labelled "may". In that case each of the following six statements holds:

1. $p \sqsubseteq_M^{\text{may}} q \Rightarrow p \sqsubseteq_N^{\text{may}} q$,
2. $p =_M^{\text{may}} q \Rightarrow p =_N^{\text{may}} q$,
3. $p \sqsubseteq_M^{\text{must}} q \Rightarrow p \sqsubseteq_N^{\text{must}} q$,
4. $p =_M^{\text{must}} q \Rightarrow p =_N^{\text{must}} q$,
5. $p \sqsubseteq_M^{\text{mm}} q \Rightarrow p \sqsubseteq_N^{\text{mm}} q$,
6. $p =_M^{\text{mm}} q \Rightarrow p =_N^{\text{mm}} q$.

If there is a path labelled "may" only the first two statements hold.

In order to keep the figure lucid, most notions with η-replication have not been drawn. One should imagine that for each of the 45 different notions from finitary simulation (S^*) to divergence sensitive stability respecting delay bisimulation (DB^Δ), there is also an η-variant, only 8 of which are visible.

Theorem 1 For N and M notions of observability, each of the statements 1-6 hold if and only if indicated in Figure 2.
Proof: By the counterexamples of Figure 3, in which $=_N$ denotes both $=_N^{\text{may}}$ and $=_N^{\text{must}}$ and similarly $\neq_N$ denotes both $\neq_N^{\text{may}}$ and $\neq_N^{\text{must}}$. □

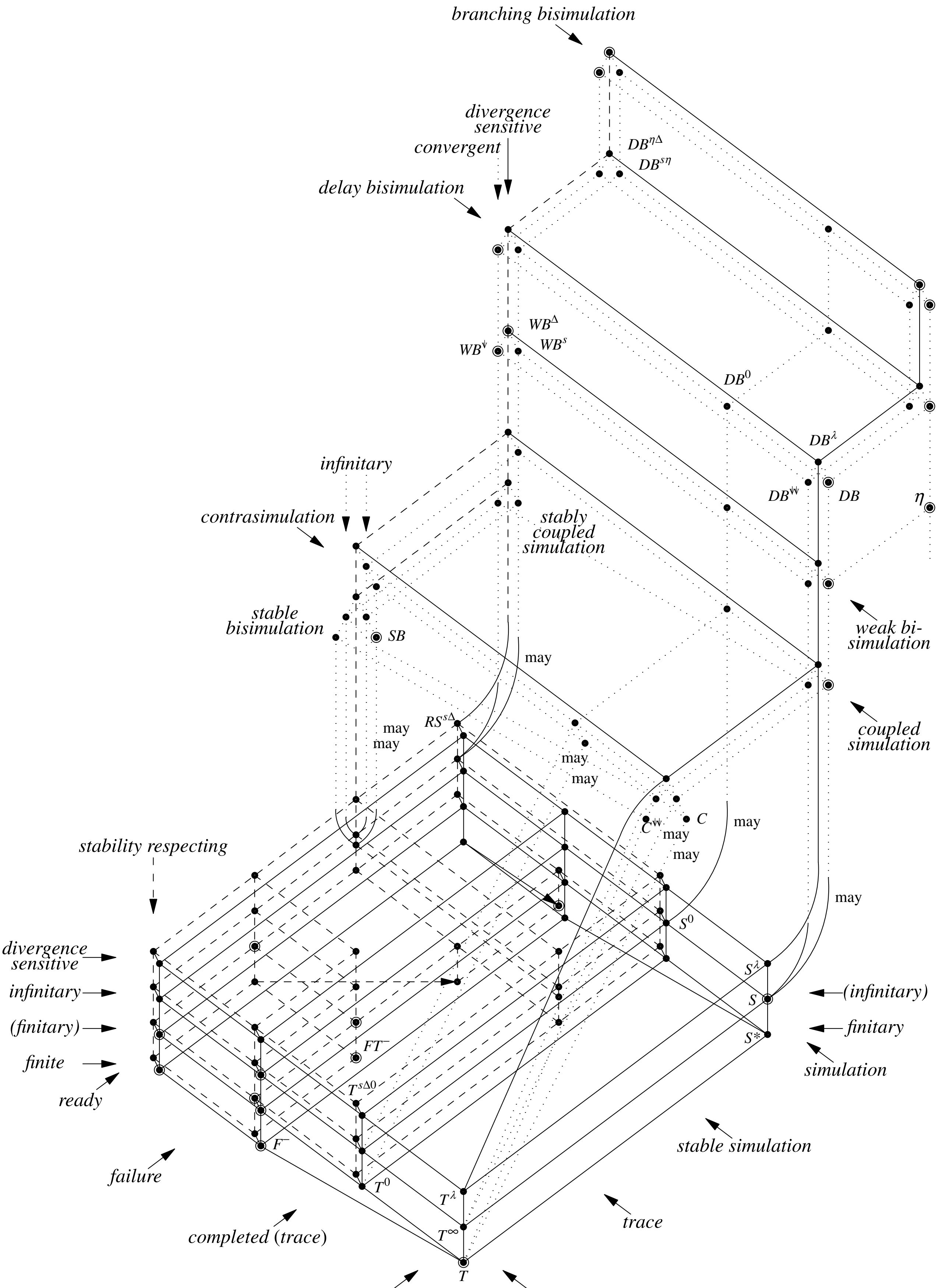

Figure 2: The linear time – branching time spectrum for infinite processes with silent moves

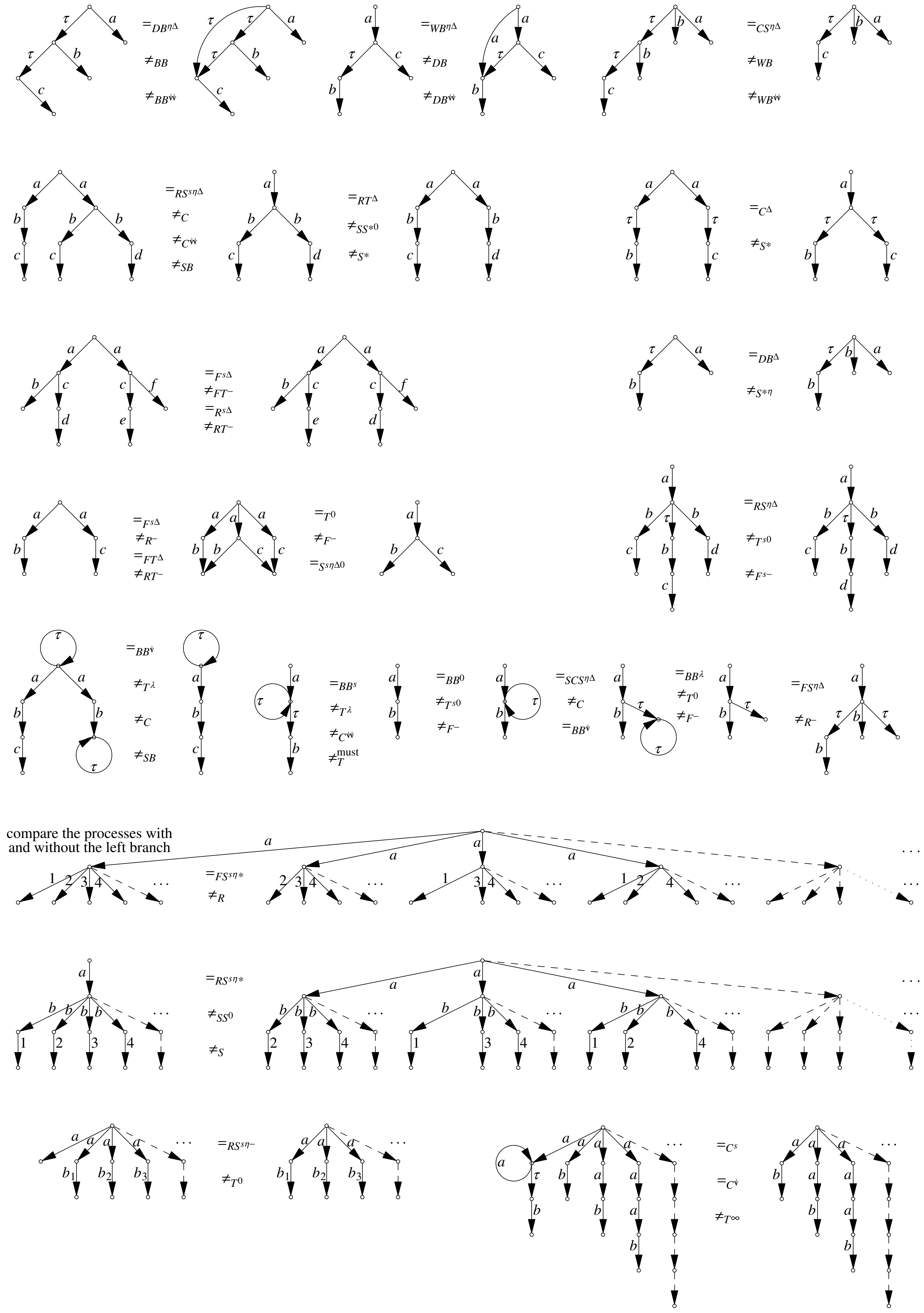

Figure 3: Counterexamples

5 Relational characterizations

Let $\mathbb{P}$ be a PLTS, labelled over $Act = A \dot{\cup} \{\tau\}$. If $p \uparrow$, the represented system may have transitions that are not explicitly present in $\mathbb{P}$. Instead of writing $p \uparrow$, this information can also be represented by drawing the hypotetical transitions in dashed lines, i.e. by letting $p \overset{\alpha}{\dashrightarrow} q$ for any $\alpha \in Act$ and $q \in \mathbb{P}$. In fact, one may wish to relax the assumption $\alpha \in Act$, and include a hypothetical action '?'. For convenience I let $\longrightarrow \subseteq \dashrightarrow$. Now just as $\Longrightarrow$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$, $\mathrel{=\!=\!\Rightarrow}$ denotes the reflexive and transitive closure of $\overset{\tau}{\dashrightarrow}$, i.e. a sequence of real and hypotetical internal transitions. In case $\not\Delta$ or $\not\lambda \in N$, one can even further extend the space of hypothetical transitions by including the ones whose presence cannot be refuted by observation.

Definition 5 Let $\mathbb{P}$ be a PLTS, labelled over Act. For $p, q \in \mathbb{P}$ and $\alpha \in Act_? = Act \dot{\cup} \{?\}$, write

- $p \overset{\alpha}{\dashrightarrow} q$ if either $p \xrightarrow{\alpha} q$ or $p \uparrow$
- $p \overset{\alpha}{\dashrightarrow}_{\not\Delta} q$ if either $p \overset{\alpha}{\dashrightarrow} q$ or $p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots$
- $p \overset{\alpha}{\dashrightarrow}_{\not\lambda} q$ if either $p \overset{\alpha}{\dashrightarrow}_{\not\Delta} q$ or $p \overset{\beta}{\not\rightarrow}$ for $\beta \in Act$.

$\mathrel{=\!=\!\Rightarrow}$ denotes the reflexive and tranistive closure of $\overset{\tau}{\dashrightarrow}$, and similarly for $\mathrel{=\!=\!\Rightarrow}_{\not\Delta}$ and $\mathrel{=\!=\!\Rightarrow}_{\not\lambda}$.
Write $p \Uparrow$ for $\exists q \in \mathbb{P} : p \Longrightarrow \overset{?}{\dashrightarrow} q$, and similarly for $p \Uparrow_{\not\Delta}$ and $p \Uparrow_{\not\lambda}$.

Definition 6 Let $(\mathbb{P}, \rightarrow, \uparrow)$ be a (κ-bounded) PLTS, labelled over $Act = A \dot{\cup} \{\tau\}$, and let N be a notion of observability containing $\bigwedge$. An N—the name of N ends on 'simulation'—is a pair of binary relations $R, S \subseteq \mathbb{P} \times \mathbb{P}$ satisfying the clauses below (in which $X \subseteq A$, $X_\tau = X \cup \{\tau\}$ and $a \in A \dot{\cup} \{?\}$). The clauses for R are the ones marked with an element of N, whereas the ones for S are the same clauses with the rôles of R and S exchanged, $\longrightarrow$ replaced by $\dashrightarrow$, $\dashrightarrow_{\not\Delta}$ or $\dashrightarrow_{\not\lambda}$, and $\Longrightarrow$ replaced by $\mathrel{=\!=\!\Rightarrow}$, $\mathrel{=\!=\!\Rightarrow}_{\not\Delta}$ or $\mathrel{=\!=\!\Rightarrow}_{\not\lambda}$, depending on the presence of $\not\Delta$ or $\not\lambda$ in N.

(a) $\quad pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q' : q \Longrightarrow \xrightarrow{a} q' \wedge p'Rq'$

$(a\epsilon)$ $\quad pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q' : q \Longrightarrow \xrightarrow{a} \Longrightarrow q' \wedge p'Rq'$

(ϵ) $\quad pRq \wedge p \xrightarrow{\tau} p' \quad \Rightarrow \quad \exists q' : q \Longrightarrow q' \wedge p'Rq'$

(0) $\quad pRq \wedge p \overset{\alpha}{\not\rightarrow} (\alpha \in A_\tau) \wedge p \downarrow \Rightarrow \exists q' : q \Longrightarrow q' \overset{\alpha}{\not\rightarrow} (\alpha \in A_\tau) \wedge q' \downarrow$

(F) $\quad pRq \wedge p \overset{\alpha}{\not\rightarrow} (\alpha \in X_\tau) \wedge p \downarrow \Rightarrow \exists q' : q \Longrightarrow q' \overset{\alpha}{\not\rightarrow} (\alpha \in X_\tau) \wedge q' \downarrow \qquad$ (similarly for F^- and $R^{(-)}$)

(FT) $pRq \wedge p \overset{\alpha}{\not\rightarrow} (\alpha \in X_\tau) \wedge p \downarrow \Rightarrow \exists q' : q \Longrightarrow q' \overset{\alpha}{\not\rightarrow} (\alpha \in X_\tau) \wedge q' \downarrow \wedge pRq'$ (similarly for FT^-, $RT^{(-)}$)

(S) $\quad pRq \wedge p \overset{\tau}{\not\rightarrow} \wedge p \downarrow \quad \Rightarrow \quad \exists q' : q \Longrightarrow q' \overset{\tau}{\not\rightarrow} \wedge q' \downarrow \wedge pRq'$

(η) $\quad pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q'', q' : q \Longrightarrow q'' \xrightarrow{a} q' \wedge pRq'' \wedge p'Rq'$

$(\eta\epsilon)$ $\quad pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q'', q' : q \Longrightarrow q'' \xrightarrow{a} \Longrightarrow q' \wedge pRq'' \wedge p'Rq'$

(b) $\quad pRq \wedge p \xrightarrow{\tau} p' \quad \Rightarrow \quad \exists q''q' : q \Longrightarrow q'' \wedge (q'' = q' \vee q'' \xrightarrow{\tau} q') \wedge pRq'' \wedge p'Rq'$

(Δ) $\quad pRq \wedge p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots \quad \Rightarrow \quad q \xrightarrow{\tau} q_1 \xrightarrow{\tau} q_2 \xrightarrow{\tau} \cdots \qquad$ (similarly for λ)

$(\neg)$ $\quad pRq \quad \Rightarrow \quad qSp$

$(\overline{\neg})$ $\quad pRq \quad \Rightarrow \quad \exists q' : q \Longrightarrow q' \wedge q'Sp$

$(\overline{\neg}_s)$ $\quad pRq \wedge p \overset{\tau}{\not\rightarrow} \wedge p \downarrow \quad \Rightarrow \quad \exists q' : q \Longrightarrow q' \overset{\tau}{\not\rightarrow} \wedge q' \downarrow \wedge q'Sp \; (\wedge pRq')$

Theorem 2 $p \sqsubseteq_N^{\text{may}} q$ iff there is an N(-simulation) (R, S) with pRq.
Similarly, $p \sqsupseteq_N^{\text{must}} q$ iff there is an N(-simulation) (R, S) with pSq.

Proof: "⇒": I show that the relations $\sqsubseteq_N^{\mathrm{may}}$ and $\sqsupseteq_N^{\mathrm{must}}$ satisfy the appropriate clauses for R and S.

(a) Suppose $\mathcal{O}(p) \subseteq \mathcal{O}(q)$ and $p \xrightarrow{a} p'$. I have to show that $\exists q' \in \mathbb{P}$ with $q \Longrightarrow\ \xrightarrow{a} q'$ and $\mathcal{O}(p') \subseteq \mathcal{O}(q')$. Suppose this is not the case. Let Q be $\{q' \in \mathbb{P} \mid q \Longrightarrow\ \xrightarrow{a} q'\}$. Then for any $q' \in Q$ there is a $\varphi_{q'} \in \mathcal{O}(p') - \mathcal{O}(q')$. Thus for any such q' one has $\bigwedge_{q' \in Q} \varphi_{q'} \in \mathcal{O}(p') - \mathcal{O}(q')$. But then $a \bigwedge_{q' \in Q} \varphi_{q'} \in \mathcal{O}(p) - \mathcal{O}(q)$, contradicting $\mathcal{O}(p) \subseteq \mathcal{O}(q)$.

(FT) Let $\mathcal{O}(p) \subseteq \mathcal{O}(q)$ and $p \downarrow\!\overset{a}{\nrightarrow}$ for $a \in X_\tau$. To show that $\exists q' \in \mathbb{P}$ with $q \Longrightarrow q'\!\downarrow$, $q' \overset{a}{\nrightarrow}$ for $a \in X_\tau$ and $\mathcal{O}(p) \subseteq \mathcal{O}(q')$. Suppose this is not the case. Let Q be $\{q' \in \mathbb{P} \mid q \Longrightarrow q'\!\downarrow \wedge q' \overset{a}{\nrightarrow}$ for $a \in X_\tau\}$. Then for any $q' \in Q$ there is a $\varphi_{q'} \in \mathcal{O}(p) - \mathcal{O}(q')$. Thus for any such q' one has $\bigwedge_{q' \in Q} \varphi_{q'} \in \mathcal{O}(p) - \mathcal{O}(q')$. But then $\widetilde{X} \bigwedge_{q' \in Q} \varphi_{q'} \in \mathcal{O}(p) - \mathcal{O}(q)$, contradicting $\mathcal{O}(p) \subseteq \mathcal{O}(q)$.

(b) Let $\mathcal{O}(p) \subseteq \mathcal{O}(q)$ and $p \xrightarrow{\tau} p'$. To show that $\exists q'', q' \in \mathbb{P}$ with $q \Longrightarrow q'' \wedge (q'' = q' \vee q'' \xrightarrow{\tau} q') \wedge \mathcal{O}(p) \subseteq \mathcal{O}(q'') \wedge \mathcal{O}(p') \subseteq \mathcal{O}(q')$. Let Q'' be $\{q'' \in \mathbb{P} \mid q \Longrightarrow q'' \wedge \exists \varphi_{q''} \in \mathcal{O}(p) - \mathcal{O}(q'')\}$ and let Q' be $\{q' \in \mathbb{P} \mid q \Longrightarrow q' \wedge \exists \varphi_{q'} \in \mathcal{O}(p') - \mathcal{O}(q')\}$. Then $(\bigwedge_{q'' \in Q''} \varphi_{q''})\tau(\bigwedge_{q' \in Q'} \varphi_{q'}) \in \mathcal{O}(p) \subseteq \mathcal{O}(q)$, so there must be $q'', q' \in \mathbb{P}$ with $q \Longrightarrow q'' \wedge (q'' = q' \vee q'' \xrightarrow{\tau} q')$, such that $q'' \notin Q''$ and $q' \notin Q'$.

($\neg$) Suppose $\mathcal{O}(p) \subseteq \mathcal{O}(q)$ and $\varphi \notin \mathcal{H}(p)$. Then $\neg\varphi \in \mathcal{O}(p) \subseteq \mathcal{O}(q)$, and this can only be the case if $\varphi \notin \mathcal{H}(q)$. Hence $\mathcal{H}(q) \subseteq \mathcal{H}(p)$, which had to be proved.

($\bar{\neg}$) Let $\mathcal{O}(p) \subseteq \mathcal{O}(q)$. I have to show that $\exists q' \in \mathbb{P}$ with $q \Longrightarrow q'$ and $\mathcal{H}(q') \subseteq \mathcal{H}(p)$. Let Q be $\{q' \mid q \Longrightarrow q' \wedge \exists \varphi_{q'} \in \mathcal{H}(q') - \mathcal{H}(p)\}$. Then $\bar{\neg} \bigvee_{q' \in Q} \varphi_{q'} \in \mathcal{O}(p) \subseteq \mathcal{O}(q)$, so $\bigvee_{q' \in Q} \varphi_{q'} \notin \mathcal{O}(q'')$ for certain $q'' \in Q$ with $q \Longrightarrow q''$, and hence $q'' \notin Q$, which had to be proved.

The other cases for $\sqsubseteq_N^{\mathrm{may}}$ are trivial or similar. The cases for $\sqsupseteq_N^{\mathrm{must}}$ follow the same lines, but the proofs have to be slightly adapted to deal with divergence. I present one representative case. In case $\not\Delta$ or $\not\lambda \in N$, the appropriate subscripts should be added to $\overset{a}{\dashrightarrow}$, $=\!=\!\Rightarrow$ and $\Uparrow$.

($a\epsilon$) Suppose $\mathcal{H}(p) \subseteq \mathcal{H}(q)$ and $p \overset{a}{\dashrightarrow} p'$. I have to show that $\exists q' \in \mathbb{P}$ with $q =\!=\!\Rightarrow\ \overset{a}{\dashrightarrow}\ =\!=\!\Rightarrow q'$ and $\mathcal{H}(p') \subseteq \mathcal{H}(q')$. There are two possibilities: $p \Uparrow$ or $a \in A$ and $p \xrightarrow{a} p'$.

In the former case $\epsilon\bot \in \mathcal{H}(p) \subseteq \mathcal{H}(q)$, where $\bot$ is the empty disjunction, or any other observation we can be sure of that it can not be made for any process in $\mathbb{P}$. But $\epsilon\bot \in \mathcal{H}(q)$ is only possible if $q \Uparrow$, and then trivially $\exists q' \in \mathbb{P}$ with $q =\!=\!\Rightarrow\ \overset{a}{\dashrightarrow}\ =\!=\!\Rightarrow q'$ and $\mathcal{H}(p') \subseteq \mathcal{H}(q')$.

In the latter case let Q be $\{q'' \in \mathbb{P} \mid q \Longrightarrow\ \xrightarrow{a}\ \Longrightarrow q''\}$. If either $q \Uparrow$ or $q'' \Uparrow$ for certain $q'' \in Q$, a $q' \in \mathbb{P}$ as required trivially exists. Suppose however that there is no such q'. Then for any $q'' \in Q$, there is a $\varphi_{q''} \in \mathcal{O}(p') - \mathcal{O}(q'')$ and hence $\bigwedge_{q'' \in Q} \varphi_{q''} \in \mathcal{O}(p') - \mathcal{O}(q'')$. But, using that neither $q \Uparrow$ nor $q'' \Uparrow$, $a\epsilon \bigwedge_{q'' \in Q} \varphi_{q''} \in \mathcal{O}(p) - \mathcal{O}(q)$, contradicting $\mathcal{O}(p) \subseteq \mathcal{O}(q)$.

"⇐": Let (R, S) be a pair of relations that satisfy the appropriate clauses. I have to show that $pRq \Rightarrow (\varphi \in \mathcal{O}_N(p) \Rightarrow \varphi \in \mathcal{O}_N(q))$ and $pSq \Rightarrow (\varphi \in \mathcal{H}_N(p) \Rightarrow \varphi \in \mathcal{H}_N(q))$ for $\varphi \in \mathbb{O}_N$. I will do so with induction on φ. First note that always a or $a\epsilon \in N$ and $\epsilon \in N$. Hence $pSq \wedge p \Uparrow \Rightarrow\ q \Uparrow$.

(a) Suppose pRq and $a\varphi \in \mathcal{O}(p)$. Then there is a $p' \in \mathbb{P}$ with $p \Longrightarrow\ \xrightarrow{a} p'$ and $\varphi \in \mathcal{O}(p')$. As R satisfies clauses (a) and (ϵ), there must be a $q' \in \mathbb{P}$ with $q \Longrightarrow\ \xrightarrow{a} q'$ and $p'Rq'$. So by induction $\varphi \in \mathcal{O}(q')$, and hence $a\varphi \in \mathcal{O}(q)$.

Suppose pSq and $a\varphi \in \mathcal{H}(p)$. Then $p \Uparrow$ or there is a $p' \in \mathbb{P}$ with $p \Longrightarrow\ \xrightarrow{a} p'$ and $\varphi \in \mathcal{H}(p')$. In the former case $q \Uparrow$ and hence $a\varphi \in \mathcal{H}(q)$. In the latter case, since S satisfies clauses (a) and (ϵ), there must be a $q' \in \mathbb{P}$ with $q =\!=\!\Rightarrow\ \overset{a}{\dashrightarrow} q'$ and $p'Sq'$. So by induction $\varphi \in \mathcal{H}(q')$, and hence $a\varphi \in \mathcal{H}(q)$ by applying clauses (a), (τ) and possibly ($\uparrow, \not\Delta, \not\lambda$) of Definition 3.

($\bar{\neg}_s$) Suppose pSq and $\bar{\neg}_s\varphi \in \mathcal{H}(p)$. Then $p \Uparrow$ or there is a $p' \in \mathbb{P}$ with $p \Longrightarrow p' \downarrow\!\overset{\tau}{\nrightarrow}$ and $\varphi \notin \mathcal{O}(p')$. In the former case $q \Uparrow$ and hence $\bar{\neg}_s\varphi \in \mathcal{H}(q)$. In the latter case, since S satisfies ($\bar{\neg}_s$) and (ϵ), $\exists q' \in \mathbb{P}$ with $q =\!=\!\Rightarrow q' \downarrow\!\overset{\tau}{\nrightarrow}$ and $q'Rp'$. So by induction $\varphi \notin \mathcal{O}(q')$, and hence $\bar{\neg}_s\varphi \in \mathcal{H}(q)$.

The other cases go likewise. □

For $\sigma \in A^*$, say $\sigma = a_1 a_2 \cdots a_n$, $p \overset{\sigma}{\Longrightarrow} q$ denotes $p \Longrightarrow \xrightarrow{a_1} \Longrightarrow \xrightarrow{a_2} \Longrightarrow \cdots \Longrightarrow \xrightarrow{a_n} \Longrightarrow q$. For σ a sequence over $A_? = A \cup \{?\}$, the relations $\overset{\sigma}{\Longrightarrow}$, $\overset{\sigma}{\Longrightarrow}_{\not\Delta}$ and $\overset{\sigma}{\Longrightarrow}_{\not\chi}$ are defined likewise. These *multiple-action relations* allow for alternative formulations of some of the relations of Definition 6, in which the clauses (a) and (ϵ) are merged into one. Besides, they allow a relational characterization of the various contra- and stable simulation preorders, that have no relational characterization in terms of single-action relations. For $\alpha \in Act_?$ define $\hat{\alpha} \in A_?^*$ by $\hat{a} = a$ for $a \in A_?$ and $\hat{\tau} = \epsilon$.

Definition 7 Let $(\mathbb{P}, \rightarrow, \uparrow)$ be a PLTS, labelled over $Act = A \,\dot{\cup}\, \{\tau\}$, and let N be a notion of observability containing $\bar{\neg}$, $\bigwedge_{s}$ or $\bigwedge$, not η, and not $\neg$ with a. An N is a pair of binary relations $R, S \subseteq \mathbb{P} \times \mathbb{P}$ satisfying the clauses below (applying the same conventions as before).

$$(\bigwedge) \quad pRq \wedge p \overset{\sigma}{\Longrightarrow} p' \quad \Rightarrow \quad \exists q' : q \overset{\sigma}{\Longrightarrow} q' \wedge p'Rq'$$

$$(\bigwedge_{s}) \quad pRq \wedge p \overset{\sigma}{\Longrightarrow} p' \downarrow \overset{\tau}{\not\longrightarrow} \quad \Rightarrow \quad \exists q' : q \overset{\sigma}{\Longrightarrow} q' \downarrow \overset{\tau}{\not\longrightarrow} \wedge p'Rq'$$

$$(T) \quad pRq \wedge p \overset{\sigma}{\Longrightarrow} p' \quad \Rightarrow \quad \exists q' : q \overset{\sigma}{\Longrightarrow} q'$$

$$(\bar{\neg}) \quad pRq \wedge p \overset{\sigma}{\Longrightarrow} p' \quad \Rightarrow \quad \exists q' : q \overset{\sigma}{\Longrightarrow} q' \wedge q'Sp$$

Clauses (0), $(F) \ldots (RT^-)$, (S), (Δ), (λ), $(\neg)$ and $(\bar{\neg}_s)$ are exactly as in in Definition 6, and clause (∞) is a more complicated version of clause (T).

Note that Definitions 6 and 7 agree where they define the same notions.

Theorem 3 $p \sqsubseteq_N^{\text{may}} q$ iff there is an N(-simulation) (R, S) with pRq.
Similarly, $p \sqsupseteq_N^{\text{must}} q$ iff there is an N(-simulation) (R, S) with pSq.

Proof: "$\Rightarrow$" goes just as in the previous proof. "$\Leftarrow$" goes again with induction on φ, but is a bit more complicated then in the previous proof because the case $\varphi = a\psi$ does not work. However, every observation φ that is not obtained by clause (∞) is of the form $\sigma\varphi'$ with $\sigma \in A^*$ and φ' not of the form $a\psi$. Now the inductive proof lumps σ together with the next operator, so that for instance the case $\sigma\bar{\neg}\varphi$ is considered, with the induction hypothesis applied to φ. This is straightforward. □

6 The literature

All encircled dots in Figure 2 correspond with preorders or equivalences found in the literature.

Trace and failure semantics The trace (may) preorder $\sqsubseteq_T$ and equivalence $=_T$, obtained by taking $N = \{T, a\epsilon, \tau, \epsilon, \bigvee\}$, is the familiar notion that can be found, for instance, in HOARE [23]. The failure must preorder $\sqsubseteq_F^{\text{must}}$, is the preorder $\sqsubseteq$ proposed in BROOKES & ROSCOE [10], a typed version of which appears in HOARE [24]. The divergences of [10, 24] correspond with observations of the form $\sigma\bot$ with $\sigma \in A^*$ and $\bot$ the empty disjunction. In BROOKES, HOARE & ROSCOE [9] a version of failure semantics appears in which the refusal sets are finite. The resulting preorder is close to what here is called the finite failure must preorder ($\sqsubseteq_{F^-}^{\text{must}}$), but the divergences were missing. As this was considered to be a shortcoming, they were added in [10].

The may and must testing preorders from HENNESSY & DE NICOLA [14] correspond with the trace may ($\sqsubseteq_T^{\text{may}}$) and failure must preorder ($\sqsubseteq_F^{\text{must}}$) in my framework. The reason for this discrepancy is that the environment in the setup of [14] is a CCS-context (of a particular form). With some effort a CCS-context can be recognized as a generative machine with switches and display, but no lights or duplication button. This explains the preorder $\sqsubseteq_T^{\text{may}}$. For must testing however, something equivalent to the green light is observable as well, due to the presence of interleaving operators. Namely put a process p (to be tested) in the context were it is interleaved with a system that performs an internal action and subsequently reports success. If p is idle, the context will perform the internal action and must report success; but as long as p is not idle, there is no certainty that the silent action of the context will ever be scheduled, so success is not guaranteed. Thus the necessity of success in this context corresponds to the green light going off in my setup, but can only occur at the end of an observation, explaining why $\sqsubseteq_F^{\rm must}$ is obtained. It should be noted that the must preorder of [14] would have been coarser if the CCS composition $(|)$ would have been interpreted as a parallel composition. In that case in would not depend on p whether the parallel component must reach the state were success is reported, and there would be no way detect idle states.

In De Nicola [13] the must equivalence of [14] has been shown to coincide with the failure equivalence of [9, 24] as well as with two equivalences proposed by Kennaway and Darondeau respectively. The results were established for finitely branching processes without divergence. The preorders of [14] were defined for a PLTS such that processes with infinitely many outgoing transitions are underspecified. On such a domain the preorders $\sqsubseteq_F^{\rm must}$ and $\sqsubseteq_{F^-}^{\rm must}$ coincide. Outside this domain both notions are slightly problematic, and in order to cope better with unbounded nondeterminism an infinitary version of failure semantics was proposed in Roscoe [37], corresponding with my infinitary failure must equivalence ($=_{F^\infty}^{\rm must}$).

In Bergstra, Klop & Olderog [7] three versions of failure semantics, among which two new ones, were compared, namely *failure equivalence with explicit divergence*, *failure equivalence with catastrophic divergence*, and *failure equivalence with fair abstraction of unstable divergence.* If I abstract from the root condition (F2), used to make the equivalence a congruence w.r.t. the $+$-operator, and the distinction between deadlock and successful termination (F3) in [7], these are $=_F^{\rm mm}$, $=_F^{\rm must}$ and $=_F^{\rm may}$ respectively.

Other decorated trace semantics Readiness semantics and the preorder $\sqsubseteq_R^{\rm must}$ comes from Olderog & Hoare [30]. Finite readiness semantics ($=_{R^-}$) is studied in Rounds & Brookes [38] under the name *acceptance-refusal semantics*, but only for processes without silent steps. The preorders $\sqsubseteq_{FT^-}^{\rm may}$, $\sqsubseteq_{FT^-}^{\rm must}$ and $\sqsubseteq_{FT^-}^{\rm mm}$ originate from Phillips [34]. There $=_{F^-}^{\rm mm}$ is called *refusal equivalence*. Ready trace equivalence ($=_{RT}$) was introduced independently by Pomello [36] (as *exhibited behaviour equivalence*), Pnueli [35] (as *barbed equivalence*), and Baeten, Bergstra & Klop [4]. In [36] the menus are collected also in non-stable states, resulting in an equivalence that falls outside my classification, but coincides with $=_{RT}$ for τ-free processes. The versions of [35] and [4] are defined for τ-free processes only. The generalization to processes with silent moves, as well as the analogous notion of failure trace equivalence ($=_{FT}$), was first reported in Baeten [2], who refers further back to the research reported here. Failure equivalence and the failure may preorder ($\sqsubseteq_{FT}$) also appear in Langerak [26], where moreover a *generalized failure* preorder and equivalence is introduced, corresponding with my stability respecting failure preorder ($\sqsubseteq_{F^s}$) and equivalence.

Simulation and ready simulation semantics The simulation preorder ($\sqsubseteq_S$) for systems without invisible actions appears in Park [31]. In Bloom, Istrail & Meyer [8] the ready simulation preorder ($\sqsubseteq_{RS}$) is introduced for finitely branching processes without silent moves. Several arguments are presented suggesting that $=_{RS}$ is the finest equivalence that 'makes computationally meaningful distinctions'. In accordance with this analysis, all finer equivalences that appear here use some form of global testing. In Ulidowski [40] the work of [8] is generalized to a setting with silent moves. Ulidowski concludes that *copy+refusal equivalence*, attributed to I. Phillips [*Copy testing*, unpublished manuscript, 1985], is the 'finest observable equivalence'. Here copy+refusal equivalence is with what I call stability respecting finite failure simulation may-and-must equivalence ($=_{FS^s-}^{\rm mm}$). Also the corresponding may and must preorders are investigated. The present paper contributes a dozen alternative generalizations of ready simulation. The relative merits of these seem to need further study.

Weak bisimulation semantics By Definition 7 a *weak bisimulation* is a pair of relations $R, S \subseteq \mathbb{P} \times \mathbb{P}$ satisfying clauses $(\bigwedge)$ and $(\neg)$. Clause $(\neg)$ says that S is the inverse of R, so the notion can be reformulated as a binary relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying, for $\sigma \in A_?^*$,

- $pRq \wedge p \overset{\sigma}{\Longrightarrow} p' \quad \Rightarrow \quad \exists q' : q \overset{\sigma}{\Longrightarrow} q' \wedge p'Rq'$,
- $pRq \wedge q \overset{\sigma}{=\!=\!\Rightarrow} q' \quad \Rightarrow \quad \exists p' : p \overset{\sigma}{=\!=\!\Rightarrow} p' \wedge p'Rq'$.

If one restricts attention to the case that $\mathbb{P}$ is a total LTS, i.e. $\uparrow = \emptyset$, there is no difference between the relations $\overset{\sigma}{=\!=\!\Rightarrow}$ and $\overset{\sigma}{\Longrightarrow}$, and one obtains the familiar definition of a weak bisimulation from MILNER [28]. The equivalent definition of a weak bisimulation in terms of single-action relations, that appears in MILNER [29], is obtained in same way by taking clauses $(a\epsilon)$, (ϵ) and $(\neg)$ of Definition 6.

Note that in the presence of clause $(\neg)$ there is no difference between the may and must preorders. Moreover, when $\uparrow = \emptyset$ and the closure properties $\not\Delta$ and $\not\lambda$ are absent, the must preorder is the inverse of the may preorder. As in the case of weak bisimulation on a total LTS both are true, the weak bisimulation preorder coincides with the associated equivalence. Weak bisimulation equivalence is what Milner calls *observation equivalence.*

A convergent weak bisimulation ($WB^{\downarrow} = \{T, a\epsilon, \tau, \epsilon, \bigvee, 0, \bigwedge, \neg, \not\Delta\}$) is a pair of relations $R, S \subseteq \mathbb{P} \times \mathbb{P}$ satisfying clauses $(a\epsilon)$, (ϵ) and $(\neg)$ of Definition 6, with the dashed arrows subscripted by $\not\Delta$. In this context the clauses (0), (S), (F), (RT), etc. are redundant by Proposition 3. Again S is the inverse of R and the notion can be reformulated as a relation R satisfying, for $\alpha \in Act_?$,

- $pRq \wedge p \xrightarrow{\alpha} p' \;\Rightarrow\; \exists q' : q \overset{\hat\alpha}{\Longrightarrow} q' \wedge p'Rq'$,
- $pRq \wedge q \overset{\alpha}{\dashrightarrow}_{\not\Delta} q' \;\Rightarrow\; \exists p' : p \overset{\hat\alpha}{=\!=\!\Rightarrow}_{\not\Delta} p' \wedge p'Rq'$.

If $p \Uparrow_{\not\Delta} \alpha$ denotes $p \Uparrow_{\not\Delta} \vee p \overset{\hat\alpha}{\Longrightarrow} p' \Uparrow_{\not\Delta}$, and $p \Downarrow_{\not\Delta} \alpha$ denies $p \Uparrow_{\not\Delta} \alpha$, then the last clause is equivalent to

- $pRq \wedge q \overset{?}{\dashrightarrow}_{\not\Delta} \;\Rightarrow\; p \Uparrow_{\not\Delta} \alpha$ (in which $q \overset{?}{\dashrightarrow}_{\not\Delta}$ may be replaced by $q \Uparrow_{\not\Delta}$ or even $q \Uparrow_{\not\Delta} \alpha$),
- $pRq \wedge q \xrightarrow{\alpha} q' \;\Rightarrow\; p \Uparrow_{\not\Delta} \alpha \vee \exists p' : p \overset{\hat\alpha}{\Longrightarrow} p' \wedge p'Rq'$,

which is in turn equivalent to

- $pRq \wedge p \Downarrow_{\not\Delta} \alpha \;\Rightarrow\; (q \Downarrow_{\not\Delta} \alpha \wedge (q \xrightarrow{\alpha} q' \;\Rightarrow\; \exists p' : p \overset{\hat\alpha}{\Longrightarrow} p' \wedge p'Rq'))$.

In this shape the convergent weak bisimulation preorder ($\sqsubseteq_{WB^{\downarrow}}$, no difference between may and must) can be recognized as the bisimulation preorder studied in STIRLING [39], ABRAMSKY [1] and WALKER [41], the idea of which originates from HENNESSY & PLOTKIN [22].

In BERGSTRA, KLOP & OLDEROG [7] *bisimulation semantics with explicit divergence* is introduced. If I abstract from the root condition (B4) and the distinction between deadlock and successful termination (B5) in [7], their Δ-*bisimulation*—the Δ is pronounced 'delay' and is used as a constant representing divergence in their system description language—is a symmetric relation satisfying properties $(\bigwedge)$ and (Δ). Thus the resulting equivalence is what I call divergence sensitive stability respecting weak bisimulation equivalence ($=_{WB^{\Delta}}$). Note that WB^{Δ} is finer than both WB and $WB^{\downarrow}$, which are incomparable.

Delay bisimulation semantics A convergent delay bisimulation $DB^{\downarrow} = \{T, a, \tau, \epsilon, \bigvee, 0, \bigwedge, \neg, \not\Delta\}$ can be formulated as a relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying, for $\alpha \in Act_?$,

- $pRq \wedge p \xrightarrow{\alpha} p' \;\Rightarrow\; \exists q' : q \overset{\hat\alpha}{\Longrightarrow}_0 q' \wedge p'Rq'$,
- $pRq \wedge q \overset{\alpha}{\dashrightarrow}_{\not\Delta}, q' \;\Rightarrow\; \exists p' : p \overset{\hat\alpha}{=\!=\!\Rightarrow}_{0\not\Delta} p' \wedge p'Rq'$.

Here $p \overset{\sigma}{\Longrightarrow}_0 q$ with $\sigma = a_1 a_2 \cdots a_n$ denotes $p \Longrightarrow \xrightarrow{a_1} \Longrightarrow \xrightarrow{a_2} \cdots \Longrightarrow \xrightarrow{a_n} q$, and similarly for $\overset{\sigma}{=\!=\!\Rightarrow}_{0\not\Delta}$. As before, the second clause is equivalent to

- $pRq \wedge p \Downarrow_{\not\Delta} \;\Rightarrow\; (q \Downarrow_{\not\Delta} \wedge (q \xrightarrow{a} q' \Rightarrow \exists p' : p \overset{\hat a}{\Longrightarrow}_0 p' \wedge p'Rq'))$.

This is the preorder proposed in MILNER [27], and also studied in ABRAMSKY [1] and WALKER [41]. The uniform characterization with the dashed arrows of the last two preorders may be helpful to explain why, as observed in [1, 41], the convergent weak bisimulation preorder needs the parameterized convergence predicates $\{\Downarrow_{\not\Delta} \alpha \mid \alpha \in Act\}$, whereas the convergent delay bisimulation preorder can be conveniently described by means of the single convergence predicate $\Downarrow_{\not\Delta}$.

Delay bisimulation equivalence ($=_{DB}$) stems from WEIJLAND [42]. There is was called that way because it corresponded with the Δ-bisimulation of [7] when translating asynchrony in synchrony. The name could be alternatively explained by the double arrow on right in clause (a), saying that when a button is put under pressure, we take for granted that there is a delay before it goes down. However, most other notions considered here have even more delay, so *alert bisimulation* is maybe a better name.

Branching bisimulation semantics By Definition 6 a branching bisimulation (BB) on a total LTS $\mathbb{P}$ can be formulated as a symmetric relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying, for $\alpha \in Act_{?}$,

- $pRq \wedge p \xrightarrow{\alpha} p' \quad \Rightarrow \quad \exists q''q' : q \Longrightarrow q'' \wedge ((\alpha = \tau \wedge q'' = q') \vee q'' \xrightarrow{\alpha} q') \wedge pRq'' \wedge p'Rq'$.

This is what was called a *semi branching bisimulation* in Van Glabbeek & Weijland [20], were branching bisimulation equivalence was introduced. By (the proof of) Lemma 1.1 in [20] two processes are related by a branching bisimulation iff they are related by a semi branching bisimulation, and hence the notion of branching bisimulation equivalence ($=_{BB}$) that arises here is the same as the one from [20]. In De Nicola & Vaandrager [15] a modal characterization (among two others) of branching bisimulation is provided using *until operators.* They have a binary until operator $\langle\hat{\alpha}\rangle$ for $\alpha \in Act$ defined by $\varphi\langle\hat{\alpha}\rangle\psi \notin \mathbb{O}(\tau)$ and

$$\varphi\langle\hat{\alpha}\rangle\psi \in \mathcal{O}(p) \text{ iff } \alpha{=}\tau \wedge \psi \in \mathcal{O}(p) \text{ or } p = p_0 \xrightarrow{\tau} \cdots \xrightarrow{\tau} p_n \xrightarrow{\alpha} q \wedge \varphi \in \mathcal{O}(p_i)\ (i \leq n) \wedge \psi \in \mathcal{O}(q).$$

Alternatively they could have used the operators $\langle\alpha\rangle$, defined similarly, but with $\alpha{=}\tau \wedge \varphi, \psi \in \mathcal{O}(p)$ instead of $\alpha{=}\tau \wedge \psi \in \mathcal{O}(p)$, which are equally expressive since

$$\varphi\langle\epsilon\rangle\psi \in \mathcal{O}(p) \quad \Leftrightarrow \quad \psi \vee \varphi\langle\tau\rangle\psi \in \mathcal{O}(p) \qquad \text{and} \qquad \varphi\langle\tau\rangle\psi \in \mathcal{O}(p) \quad \Leftrightarrow \quad \varphi \wedge \varphi\langle\epsilon\rangle\psi \in \mathcal{O}(p).$$

The operators $\langle\hat{\alpha}\rangle$ and $\langle\alpha\rangle$ clearly grasp the idea of continuous copying. But in the presence of $\neg$ and $\bigvee$, the apparently weaker operators η and b—weaker because $\varphi\alpha\psi \Leftrightarrow \epsilon(\varphi\langle\alpha\rangle\psi)$—induce the same identifications on PLTSs, namely branching bisimulation equivalence.

In branching bisimulation semantics the observation Δ of divergence is arguably not powerful enough. Namely the facility of continuous copying allows one to make observations of the form $\Delta\varphi$, saying that no activity is observed during an infinite period of time, but each copy made during this period admits the observation φ. Formally

$$\Delta\varphi \in \mathcal{O}_N(p) \text{ iff } p \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots \wedge \varphi \in \mathcal{O}(p_i) \text{ for } i \in \mathbb{N}.$$

The corresponding clause in the relational characterization of Definition 6 is

$$p_0 \xrightarrow{\tau} p_1 \xrightarrow{\tau} p_2 \xrightarrow{\tau} \cdots \wedge p_iRq_0 \text{ for } i \in \mathbb{N} \quad \Rightarrow \quad q_0 \xrightarrow{\tau} q_1 \xrightarrow{\tau} q_2 \xrightarrow{\tau} \cdots \wedge p_iRq_j \text{ for } i,j \in \mathbb{N}.$$

This improved notion of divergence sensitive stability respecting branching bisimulation corresponds with the *branching bisimulation with explicit divergence* of [20]. The similarly improved divergence sensitive branching bisimulation coincides, for finite state LTSs, with the one proposed in De Nicola & Vaandrager [15], that in turn coincides, after proper translation between state-labelled and transition-labelled transition systems [15], with with the *stuttering equivalence* of Browne, Clarke & Grümberg [11], that is characterized by the temporal logics CTL and CTL*, each without next-time operator.

Other bisimulation equivalences (Weak) η-bisimulation equivalence ($=_{WB^\eta}$), originates from Baeten & Van Glabbeek [5], the preorders $\sqsubseteq_{WB^{\eta\downarrow}}$ and $\sqsubseteq_{BB^\downarrow}$ are mentioned in [20], and the equivalence $=_{DB^\eta}$ is introduced independently as *quasi branching bisimulation equivalence* in Cherief [12] and as *simple branching bisimulation* by Jan van Eijck [personal communication].

Coupled simulation and stable bisimulation semantics Stable bisimulation semantics was introduced in Van Glabbeek [16] for finite processes (representable as finite trees without divergence). Its name and complete axiomatization was quoted already in Beaten [2], but there the definition was left as an exercise for the reader. The present paper offers 13 possible generalizations (including the contrasimulations) of this notion to infinite processes with divergence. Similarly, coupled simulation for divergence-free processes was introduced in Parrow & Sjödin [32]. The present paper presents offers 9 generalizations to processes with divergence. One of those, in my terminology simply 'coupled simulation', is called *weakly coupled simulation* in Parrow & Sjödin [33], where for divergence-free agents it is proven to coincide with the notion from [32].

Possible futures and nested simulations The notions of *n-nested simulation equivalence* of Groote & Vaandrager [21] and *possible futures* equivalence of Rounds & Brookes [38] have never been considered in a setting with silent moves. However, the former can be treated by considering the observations for weak or stable bisimulation that have at most $n-1$ levels of nesting of occurrences of the negation operator, and the latter by considering the observations for (stable) simulation that have no nested occurrences of the conjunction.

7 Concluding remarks

In this paper 155 notions of observability and corresponding must and may equivalences have been reviewed. The reason for selecting just these, is that they can be motivated rather nicely with a testing scenario and/or play a rôle in the literature on semantic equivalences. This paper did not exhaust all interesting equivalences. In Van Glabbeek [16] for instance I propose a *τ-spectrum*, such that an interesting equivalence is determined by a position in the linear time – branching time spectrum (Figure 2), together with a position in the τ-spectrum. This paper dealt with the most abstract point of the τ-spectrum, the most concrete being the one where τ-actions are treated just like visible actions. In between there is for instance the possibility that (roughly) corresponds to a version of Definition 3 in which $\mathbb{O}(\tau)$ only contains the observations of the form $\epsilon\varphi$.

In this paper the set A of visible actions has been fixed. This means that a minimum level of abstraction is considered. It is still possible to represent processes at a higher level of abstraction, by replacing certain actions by τ. As it seems to be reasonable to require that a move towards a higher level of abstraction would not make two equivalent processes inequivalent, a minimum requirement for a useful equivalence could be that it is a congruence for the *abstraction operator* (remaining into τ). The 21 'stability respecting' notions that contain observation S, but not FT (or the stronger observation $\underset{S}{\neg}$) (including the one from [26]) fail this requirement, as can be seen by renaming f into τ in the only counterexample of Figure 3 that features an f, and by abstracting from c in the example below. Moreover, the 'finite' notions (with a $^-$, i.e. the bottom plane in Figure 2) are not preserved under abstraction of infinitely many actions (namely the actions b_i in a version of the appropriate example in Figure 3 with a τ-loop after each b_i). The 'finite' preorders and equivalences also fail to be preserved under other infinitary non-injective relabellings (renaming infinitely many actions into the same action $a \in A$). This may explain why such relabellings are excluded in the system description language CSP. It does not contradict the congruence theorem in Phillips [34], as that refers to a version of CCS with only injective relabellings. At several places in the literature versions of failure or readiness semantics appear that take refusal sets or menus also in unstable states. For $\sqsubseteq_F^{\rm must}$ this doesn't make a difference, but elsewhere this leads again to preorders that are not preserved under abstraction. For this reason these approaches are not treated here. Also notions with explicit divergence that do not contain infinitely long observations yield no congruences for abstraction (except of course when the infinite observations can be deduced from the finite ones, as is the case for finite-state behaviours), and are therefore not considered here. Finitary must preorders are precongruences for abstraction when restricted to systems that diverge where they are infinitely branching (as in [14]). If infinitary abstraction was considered in CSP [9, 10, 24], where this restriction is not observed, it could not commute with finitary abstraction (i.e. it matters whether first a single actions is hidden and subsequently an infinite set of actions or vice versa). It is for this reason that an infinitary version of failure semantics was proposed in Roscoe [37]. However, infinitary equivalences without $\bigwedge$ or $\bar{\neg}, \neg$ fail to satisfy the Recursive Specification Principle (RSP) of Baeten, Bergstra & Klop [3], that says that systems of guarded recursive equations have a unique solution.

At this place I can only touch upon the criteria that could be applied for selecting one equivalence over the other. Being a congruence for abstraction is one, and being a congruence for the other CSP operators could be another such criterion. This would rule out the 16 'completed' may and must preorders, which fail to be precongruences for the restriction operator of CCS or the synchronous composition of CSP. I conjecture that the $155-21-16=118$ remaining may and must preorders are preserved by all CSP operators, and the $118-11=107$ non-finite ones also for the CCS operators except $+$. Another criterion could be satisfying RSP. This would leave $107-15=92$ notions.

It is often argued that global testing is not a realistic testing scenario, and therefore some variant of ready simulation semantics represents the limit of observable behaviour. Obviously, testing for divergence is also unrealistic as it requires infinitely long observations. However, since LAMPORT [25], there is widespread consensus that *lifeness* properties are important in the verification of concurrent systems. This means (among others) that one wants to distinguish between a system $a\tau b$ that surely will do a b some day, and a system $a(\tau b+\tau^\omega)$ that very well may fail to ever do a b-move. This distinction can be made with global testing or explicit divergence, but also with must-preorders, which are (therefore?) very popular, even omong the ones that reject global testing (such as ULIDOWSKI [40]). However, the difference between $a\tau b$ and $a(\tau b+\tau^\omega)$ is not observable by finitary means without assuming some form of global testing. I therefore argue that must-testing is a special case of global testing and if one accepts must-testing, the case against general global testing is very weak. In each case something fairly unobservable is needed to cope with lifeness. Furthermore, if one rejects the progress assumption made earlier, lifeness is lost from the onset, and also the concept of must-testing becomes meaningless.

Koomen's Fair Abstraction Rule (KFAR) (BAETEN, BERGSTRA & KLOP [3]), expresses a global fairness assumption. It says that when possible a system will escape from any cycle of internal actions. Some form of KFAR is crucial for many protocal verifications with unreliable channels, and for that reason preorders and equivalences that satisfy KFAR are of special interest. Must preorders and divergence sensitive ones cannot satisfy KFAR. In BERGSTRA, KLOP & OLDEROG [7] it is shown that the combination of KFAR with failure semantics is inconsistent, but they formulate a weaker version of KFAR that is satisfied in failure may-semantics. Still the combination of KFAR$^-$ and the liveness requirement appears to require global testing, and is only satisfied in the semantics between contrasimulation (C) and stability respecting branching bisimulation (BB^s). These requirements would reduce the number of suitable preorders to 18.

It is in general a good strategy to do your verifications using the finest preorder possible. This way you don't have to redo your verifications if you decide that your original preorder was not fine enough (but as verifications deal usual with equational properties, they remain valid if you descend to a coarser preorder [19]). This is in my opinion the most compelling argument for using bisimulation semantics, and branching bisimulation in particular [20]. But sometimes a verification cannot be carried out in bisimulation semantics, as the systems compared are not (weak or branching) bisimulation equivalent. Examples of this can be found in VAN GLABBEEK & VAANDRAGER [19] and PARROW AND SJÖDIN [32]. In these cases, and probably in many others, it is not nessesary to descend to the level of for instance failure semantics; any equivalence of this paper that is coarser then weak bisimulation would do. In combination with the requirements KFAR and lifeness (which are not relevant or even desirable in every situation), this would leave just coupled simulation and contrasimulation (in a few variations). In [32], were coupled simulation was introduced, another argument was given, namely that it has a relational characterization in terms of single-action relations, which makes it easy to establish the existence of a coupled simulation between two processes. This advantage is not shared with contrasimulation. But if one would not be interested in KFAR, also the various (ready-)simulations share this property.

It is well known that most of the equivalences discussed here fail to be congruences w.r.t. the CCS $+$. One approach to this problem is to abandon this operator in favor of CSP-styled choice operators. Another approach is to refine the equivalenes slightly, only affecting the part near the initial state of processes, so that they become congruences [29]. Here, from the second point of view, I would like to argue that the problem arises from not fully observing the green light. The CCS processes 0 and $\tau.0$ for instance are equivalent under all notions considered here, but they are not congruent, as $a+0$ differs from $a+\tau.0$. But if one can not only observe that the green light is *off*, but also that it is *on*, the processes are no longer equivalent, as in $\tau.0$ the light goes on and then off, whereas in 0 it stays of. Now upgrading the notions of observability with an observation 1 (saying that the light is on), gives exactly the required congruences. In case we have $1\in N$ but not $0 \in N$, one can observe it when the light is on, but not that it is off. This idea needs futher elaboration, and yields a fourth point on the τ-spectrum.

It is left for a future occassion to extend this work with parallelism, and to establish complete axiomatizations. Complete axiomatizations (without proofs) of T, T^0, F, R, FT, RT, SB and B congruence for finite divergence-free processes are collected already in [16] and an axiomatization for coupled simulation congruence is provided in PARROW & SJÖDIN [33].

Acknowledgements

Thanks are due to Jan Willen Klop, Jan Bergstra, Frits Vaandrager, Jos Baeten, Jan Friso Groote, Ernst Olderog, Joachim Parrow, and many others, for stimulating discussions on the material of this paper.

References

[1] S. ABRAMSKY (1987): *Observation equivalence as a testing equivalence. Theoretical Computer Science* 53, pp. 225–241.

[2] J.C.M. BAETEN (1986): *Procesalgebra.* Programmatuurkunde. Kluwer, Deventer. In Dutch.

[3] J.C.M. BAETEN, J.A. BERGSTRA & J.W. KLOP (1987): *On the consistency of Koomen's fair abstraction rule. Theoretical Computer Science* 51(1/2), pp. 129–176.

[4] J.C.M. BAETEN, J.A. BERGSTRA & J.W. KLOP (1987): *Ready-trace semantics for concrete process algebra with the priority operator. Computer Journal* 30(6), pp. 498–506.

[5] J.C.M. BAETEN & R.J. VAN GLABBEEK (1987): *Another look at abstraction in process algebra.* In Th. Ottmann, editor: *Proceedings* 14th *ICALP,* Karlsruhe, *Lecture Notes in Computer Science* 267, Springer-Verlag, pp. 84–94.

[6] J.W. DE BAKKER, J.N. KOK, J.-J.CH. MEYER, E.-R. OLDEROG & J.I. ZUCKER (1986): *Contrasting themes in the semantics of imperative concurrency.* In J.W. de Bakker, W.P. de Roever & G. Rozenberg, editors: *Current Trends in Concurrency, Lecture Notes in Computer Science* 224, Springer-Verlag, pp. 51–121.

[7] J.A. BERGSTRA, J.W. KLOP & E.-R. OLDEROG (1987): *Failures without chaos: a new process semantics for fair abstraction.* In M. Wirsing, editor: *Formal Description of Programming Concepts – III, Proceedings of the* 3th *IFIP WG 2.2 working conference,* Ebberup 1986, North-Holland, Amsterdam, pp. 77–103.

[8] B. BLOOM, S. ISTRAIL & A.R. MEYER (1988): *Bisimulation can't be traced: Preliminary report.* In *Conference Record of the* 15th *ACM Symposium on Principles of Programming Languages,* San Diego, California, pp. 229–239. Full version available as Technical Report 90-1150, Department of Computer Science, Cornell University, Ithaca, New York, August 1990. Accepted to appear in *Journal of the ACM.*

[9] S.D. BROOKES, C.A.R. HOARE & A.W. ROSCOE (1984): *A theory of communicating sequential processes. Journal of the ACM* 31(3), pp. 560–599.

[10] S.D. BROOKES & A.W. ROSCOE (1985): *An improved failures model for communicating processes.* In S.D. Brookes, A.W. Roscoe & G. Winskel, editors: *Seminar on Concurrency, Lecture Notes in Computer Science* 197, Springer-Verlag, pp. 281–305.

[11] M.C. BROWNE, E.M. CLARKE & O. GRÜMBERG (1988): *Characterizing finite Kripke structures in propositional temporal logic. Theoretical Computer Science* 59(1,2), pp. 115–131.

[12] F. Cherief (1992): *Contributions à la sémantique du parallélisme: bisimulations pour le raffinement et le vrai parallélisme.* PhD thesis, Univ. Grenoble.

[13] R. De Nicola (1987): *Extensional equivalences for transition systems. Acta Informatica* 24, pp. 211–237.

[14] R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes. Theoretical Computer Science* 34, pp. 83–133.

[15] R. De Nicola & F.W. Vaandrager (1990): *Three logics for branching bisimulation (extended abstract).* In *Proceedings* 5^{th} *Annual Symposium on Logic in Computer Science,* Philadelphia, USA, IEEE Computer Society Press, pp. 118–129. Full version available as Rapporto di Ricerca SI-92/07, Dipartimento di Scienze dell'Informazione, Università degli Studi di Roma "La Sapienza", November 1992.

[16] R.J. van Glabbeek (1988): *De semantiek van eidige, sequentiële processes met interne acties.* Syllabus processemantieken, deel 2. Handwritten manuscript, in Dutch.

[17] R.J. van Glabbeek (1990): *The linear time – branching time spectrum.* In J.C.M. Baeten & J.W. Klop, editors: *Proceedings CONCUR 90,* Amsterdam, *Lecture Notes in Computer Science* 458, Springer-Verlag, pp. 278–297.

[18] R.J. van Glabbeek, S.A. Smolka, B. Steffen & C.M.N. Tofts (1990): *Reactive, generative, and stratified models of probabilistic processes.* In *Proceedings* 5^{th} *Annual Symposium on Logic in Computer Science,* Philadelphia, USA, IEEE Computer Society Press, pp. 130–141.

[19] R.J. van Glabbeek & F.W. Vaandrager (1989): *Modular specifications in process algebra – with curious queues (extended abstract).* In M. Wirsing & J.A. Bergstra, editors: *Algebraic Methods: Theory, Tools and Applications, Workshop Passau 1987, Lecture Notes in Computer Science* 394, Springer-Verlag, pp. 465–506.

[20] R.J. van Glabbeek & W.P. Weijland (1990): *Branching time and abstraction in bisimulation semantics.* Technical Report TUM-I9052, SFB-Bericht Nr. 342/29/90 A, Institut für Informatik, Technische Universität München, Munich, Germany. Extended abstract in G.X. Ritter, editor: *Information Processing 89*, Proceedings of the IFIP 11th World Computer Congress, San Fransisco, USA 1989, Elsevier Science Publishers B.V. (North-Holland), 1989, pp. 613-618.

[21] J.F. Groote & F.W. Vaandrager (October 1992): *Structured operational semantics and bisimulation as a congruence. Information and Computation* 100(2), pp. 202–260.

[22] M. Hennessy & G.D. Plotkin (1980): *A term model for CCS.* In P. Dembiński, editor: 9^{th} *Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science* 88, Springer-Verlag, pp. 261–274.

[23] C.A.R. Hoare (1980): *Communicating sequential processes.* In R.M. McKeag & A.M. Macnaghten, editors: *On the construction of programs – an advanced course,* Cambridge University Press, pp. 229–254.

[24] C.A.R. Hoare (1985): *Communicating Sequential Processes.* Prentice-Hall International, Englewood Cliffs.

[25] L. Lamport (1977): *Proving the correctness of multiprocess programs. IEEE Trans. Software Engin.* 3, pp. 125–143.

[26] R. Langerak (1989): *A testing theory for LOTOS using deadlock detection.* In E. Brinksma, G. Scollo & C.A. Vissers, editors: *Proceedings* 9^{th} *IFIP WG6.1 International Symposium on Protocol Specification, Testing, and Verification,* Enschede, The Netherlands.

[27] R. Milner (1981): *A modal characterisation of observable machine-behaviour.* In G. Astesiano & C. Bohm, editors: *Proceedings CAAP 81*, *Lecture Notes in Computer Science* 112, Springer-Verlag, pp. 25–34.

[28] R. Milner (1985): *Lectures on a calculus for communicating systems.* In S.D. Brookes, A.W. Roscoe & G. Winskel, editors: *Seminar on Concurrency*, *Lecture Notes in Computer Science* 197, Springer-Verlag, pp. 197–220.

[29] R. Milner (1990): *Operational and algebraic semantics of concurrent processes.* In J. van Leeuwen, editor: *Handbook of Theoretical Computer Science*, chapter 19, Elsevier Science Publishers B.V. (North-Holland), pp. 1201–1242. Alternatively see *Communication and Concurrency*, Prentice-Hall International, Englewood Cliffs, 1989, of which an earlier version appeared as *A Calculus of Communicating Systems*, LNCS 92, Springer-Verlag, 1980.

[30] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes.* *Acta Informatica* 23, pp. 9–66.

[31] D.M.R. Park (1981): *Concurrency and automata on infinite sequences.* In P. Deussen, editor: *5th GI Conference*, *Lecture Notes in Computer Science* 104, Springer-Verlag, pp. 167–183.

[32] J. Parrow & P. Sjödin (1992): *Multiway synchronization verified with coupled simulation.* In W.R. Cleaveland, editor: *Proceedings CONCUR 92,* Stony Brook, NY, USA, *Lecture Notes in Computer Science* 630, Springer-Verlag, pp. 518–533.

[33] J. Parrow & P. Sjödin (1993): *The complete axiomatization of cs-congruence.* This volume.

[34] I.C.C. Phillips (1987): *Refusal testing.* *Theoretical Computer Science* 50, pp. 241–284.

[35] A. Pnueli (1985): *Linear and branching structures in the semantics and logics of reactive systems.* In W. Brauer, editor: *Proceedings 12th ICALP,* Nafplion, *Lecture Notes in Computer Science* 194, Springer-Verlag, pp. 15–32.

[36] L. Pomello (1986): *Some equivalence notions for concurrent systems – An overview.* In G. Rozenberg, editor: *Advances in Petri Nets 1985*, *Lecture Notes in Computer Science* 222, Springer-Verlag, pp. 381–400.

[37] A.W. Roscoe (1988): *Unbounded nondeterminism in CSP.* In *To papers on CSP*, Technical Monograph PRG-67, Programming Research Group, Oxford University Computing Laboratory, Oxford, England, pp. 27–80.

[38] W.C. Rounds & S.D. Brookes (1981): *Possible futures, acceptances, refusals and communicating processes.* In *22th Annual Symposium on Foundations of Computer Science,* Nashville, Tennessee, IEEE, New York, pp. 140–149.

[39] C. Stirling (1987): *Modal logics for communicating systems.* *Theoretical Computer Science* 49, pp. 311–347.

[40] I. Ulidowski (1992): *Equivalences on observable processes.* In *Proceedings 7th Annual Symposium on Logic in Computer Science,* Santa Cruz, California, IEEE Computer Society Press, pp. 148–159.

[41] D.J. Walker (1990): *Bisimulation and divergence.* *Information and Computation* 85(2), pp. 202–241.

[42] W.P. Weijland (1989): *Synchrony and asynchrony in process algebra.* PhD thesis, University of Amsterdam.