Basic Research in Computer Science

BRICS RS-96-43 A. Ingólfsdóttir: Weak Semantics Based on Lighted Button Pressing Experiments

Weak Semantics Based on Lighted Button Pressing Experiments

An Alternative Characterization of the Readiness Semantics

Anna Ingólfsdóttir

BRICS Report Series **RS-96-43**

ISSN 0909-0878 **November 1996**

See back inner page for a list of recent publications in the BRICS Report Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through World Wide Web and anonymous FTP:

`http://www.brics.dk/`
`ftp://ftp.brics.dk/pub/BRICS`
This document in subdirectory `RS/96/43/`

Weak Semantics Based on Lighted Button Pressing Experiments

An Alternative Characterizations of the Readiness Semantics

Anna Ingólfsdóttir
BRICS *
Department of Computer Science, Aalborg University, Denmark†

Abstract

Imposing certain restrictions on the transition system that defines the behaviour of a process allows us to characterize the readiness semantics of [OH86] by means of black–box testing experiments, or more precisely by lighted button testing experiments [BM92]. As divergence is considered we give the semantics as a preorder, the readiness preorder, which kernel coincides with the readiness equivalence of [OH86]. This leads to a bisimulation like characterization and a modal characterization of the semantics. A concrete language, recursive free CCS without τ, is introduced, a proof system defined and it is shown to be sound and complete with respect to the readiness preorder. In the completeness proof the modal characterization plays an important role as it allows us to prove algebraicity of the preorder purely operationally.

1 Introduction

The behaviour of concurrent processes or complex systems is often given by operational semantics [Mil80]. Usually the operational semantics is based on a two level approach. The first level consists of assigning to the process a state transition graph which gives the different states the process can enter and the transitions between these together with informations about which action the process has to perform in a given state to enter another state. This level is usually modelled by a labelled transition system. As the labelled transition systems only describe the stepwise computation the process may perform, it is in general too concrete to abstract away from unimportant differences in the behaviour of two

*Basic Research in Computer Science, Centre of the Danish National Research Foundation.
†email: annai@iesd.auc.dk

processes. To overcome this problem equivalences have been introduced on top of the labelled transition systems to identify processes with the same or similar behaviour. These relations are usually based on the idea that two processes should be identified behaviourally unless an external observer can tell them apart. The precise definition of the relation then depends on how powerful the external observer is both what concerns what she can observe and how she can interact with the process being observed.

One way of identifying processes is by performing a so-called button-pressing experiment on them. In [BM92] this set up is described as follows.

> A process is thought of as a black box with an interface which is equipped with one button for each possible action it can make and possibly some other control devices. The observer presses the buttons. Either the button is locked and the experiment fails or it goes down and the process changes to another state. The observer then either has completed the experiment with success, or she may continue the experiment in the new state.

In this study we use the convention that the observer performs an experiment by applying a *test* on the process, by which we mean a description of how the button pressing is going to take place depending on how the process reacts. If the experiment fails the process is said to *fail* the test and if it is successful we say that the process *passes* the test. Because of the possibility of nondeterminism in the behaviour of the process, a process may sometimes pass a given test and sometimes not. Two processes are considered behaviourally equivalent if and only if the may pass exactly the same tests.

The outcome of a black box experiment strongly depends on what behavioural aspects of the process are immediately observable and what kinds of tests the observer has in her power. For instance the buttons of the black box interface might have lights inside them so the observer can immediately see which actions are available in a given state without pressing the buttons and thereby the process changing state. This testing scenario is referred to as *lighted button pressing experiments* in [BM92]. Alternatively there might be no lights inside the buttons in which case the observer has to press the button and possibly change the state of the process to observe the ability of performing that action (*blind button pressing experiments*). Also the tests which are allowed in the experiments can be specified in different ways.

It has become a standard practice to distinguish between external and internal actions that occur in a computation of a process. (An internal action occurs for instance when a system of two parallel processes changes state as a consequence of a private communication.) In the literature researchers distinguish between strong semantics, which allows the observer to record the internal actions, and the weak semantics where the internal actions ar not observable. The button pressing experiments in [BM92] only concerns the strong semantic approach. In

[Gla90] and [Gla93] van Glabbeek gives a very thorough study of most known behavioural preorders and equivalences both based on the strong and the weak semantic approach. However his main focus is on comparing the different relations, i. e. ordering them with respect to inclusion. In this paper we will study a version of the lighted button pressing scenario based on the idea of readiness semantics introduced by Olderog and Hoare in [OH86]. Unlike in [Gla93] our main purpose is to use the button pressing characterization to derive hopefully a rather simple theory to reason about process behaviour. The readiness semantics of [OH86] may be described as follows:

> The meaning of a process is given by a set of observations. The observation space is the set of all elements which either are of the form s or sA where s is a sequence of actions and A is a finite set of actions. We say that a process is stable if it can not perform an internal action and divergent if it can perform an infinite sequence of internal actions. The set of observations of a process, p, $obs(p)$ is defined as the least set that satisfies:
>
> - If p can perform the sequence s (possibly interspersed with internal actions) then $s \in obs(p)$,
> - if p can perform the sequence s and thereby reach a stable state p' where A is exactly the set of actions p' can perform then $sA \in obs(p)$,
> - if p diverges on s (i.e. p may perform a prefix of the string s and thereby reach a divergent state) then $obs(p)$ contains s and sA for all A.
>
> Two processes are behaviourally equivalent if their sets of observations coincide. In particular this means that the set of observations for any divergent process is equal to the whole observation space Obs. Therefore all divergent processes are identified with the inactive divergent process which in the reference mentioned above is called div.

As this semantics adopts the weak approach we have to decide how to extend the button pressing scenario described above accordingly. Here there are a few questions which have to be addressed and some assumption are to be made. Introducing internal actions into the semantics implies the possibility of divergence. How is this to be observed by an external observer? There are several possible choices but in this first approach we make the following assumption: a divergent process does not react on *any* experiment. We do not assume that the divergent process fails the experiment but rather take an intuitionistic point of view by saying that *we do not know* if it does or not. The experiment is simply aborted if the process enters a divergent state, i.e. no test terminates when applied to a divergent state. Only when the process is convergent we may say something

about the outcome of an experiment. This means in particular that failing and passing an experiment are not complementary in the sense that we may have a process where we can not decide whether it may fail or pass a given experiment. However, although divergent processes can not be tested we assume that divergence is in some sense observable. For instance we may assume a "time out" for the process to reach a stable state after a fixed period of time.

When divergence sensitive behavioural semantics is given for processes it has become standard to compare processes by a preorder rather than an equivalence. Intuitively "a process p is smaller than q" if they "have the same behaviour but p may diverge more often than q". Following this approach we say that "p is smaller than q" iff "whenever t must terminate when applied to p and p may pass t, then t must terminate on q and also q may pass t" and "whenever t must terminate when applied to p and p must fail t, then the same must hold for q".

The next assumption we want to make is that we only need to test a convergent process when it has reached a stable state. In particular this means that we assume that the process does not lose any of its communication abilities, i. e. abilities to output and input, by performing an internal action. In terms of process description languages like CCS this means that, if $p \xrightarrow{a} p'$ and $p \xrightarrow{\tau} p''$, then $p'' \xrightarrow{a} p'$, which is in general not true for this calculus. However this condition holds for the language we considered, which is the regular sub-language of the language considered in [Hen88], basically the standard CCS where the internal action τ is replaced by the internal choice operator $\oplus$. A drawback with this approach is that this condition does not hold if the parallel operator $|$ is allowed with the standard definition of the operational semantics for it. Whether the definition of the behaviour of $|$ can be modified in such a way that it fits into this set–up is an open question.

For the readiness semantics described above the restriction we impose on the labelled transition system implies that we do not have to consider separately the observations of the form s as whenever p may perform a sequence s it can perform s and thereby go to a stable state and will therefore have an observation of the form sA.

Based on the considerations above we make the following assumptions about the transition system that describes the behaviour of the process:

- Internal actions preserve the ability of performing external actions (in the sense described above).
- The process can only perform a finite number of different actions in its convergent states during its whole lifetime (weak sort finiteness).
- Each convergent state can only perform a finite number of actions, external or internal, leading to different states (weak finite branching).

Now we describe the black–box interface as follows.

- The control panel of the black-box machine has one button for each potential action of the process. The buttons are labelled with the actions and have lights inside them. It also has a red and a green control lamp. These are the only control devices.
- For all processes either the red or the green lamp will eventually light after a limited amount of time.
- If the process enters a divergent state the red lamp lights and all buttons are blocked.
- If the process converges the green lamp lights and some button lights come on as well; the experiment may start (or continue if it has already started in a previous state).

Our next task is to describe the class of tests needed in the experiments in order to characterize the readiness semantics and how the observer applies them on the process.

> The tests have the form $a_1.a_2.\cdots a_n.A, n \geq 0$ where $a_1.a_2.\ldots a_n$ is a sequence of visible or external actions and A is a finite sets of such actions. They are applied as follows: The observer waits for the control light. If it is red the experiment is aborted. If it is green, we have the following cases:
>
> - If $n = 0$ she checks if A coincides with the set of labels of the lighted buttons.
> - If $n > 0$ and a_1 is not in A she records failure, otherwise she presses the a_1-button. Then she continues by applying the remainder of the test, $a_2.\cdots a_n.A$ in the new state.

This testing scenario induces a preorder which we refer to as the readiness preorder. Not so surprisingly it turns out that the equivalence induced by this preorder is exactly the readiness equivalence given our constraints on the transition system. However this is only true if the set of all possible actions is infinite. The following example shows this.

> Let a be the only possible action. Then
>
> $$Obs = \{a^n A,\ a^n | n \geq 0 \text{ and } A = \{a\}, \emptyset\}.$$
>
> Assume furthermore that the set of states is given by $\{s_1, s_2, s_3, s_4\}$, the internal actions by $\succ\!\!\longrightarrow$ and the external ones by $\stackrel{a}{\longrightarrow}$ where
>
> $$s_1 \succ\!\!\longrightarrow s_2 \stackrel{a}{\longrightarrow} s_1, s_1 \succ\!\!\longrightarrow s_3 \text{ and } s_4 \succ\!\!\longrightarrow s_4.$$

It is easy to see that $obs(s_1) = obs(s_4) = Obs$. However $s_1 \not\sqsupseteq_{\mathcal{R}} s_4$ as $s_1 \downarrow$ and may be accepted by the test $a.\emptyset$ but $s_4 \uparrow$ and does not react on any test. Thus the readiness semantics in [OH86] does not detect divergence if the set of labels is finite.

The aim of this study is to show that this characterization of the readiness semantics leads to a rather simple bisimulation like characterization; the preorder is obtained as a least fixed point to a monotonic function on a complete lattice. We also define a modal logic to reason about process behaviour and show that it characterizes exactly the readiness preorder. Then we introduce a concrete language, $CCS_{\oplus}$, regular CCS with τ replaced by the internal choice operator $\oplus$, and give it an axiomatic semantics by means of an equationally based proof system. We prove that it is sound and complete with respect to the readiness preorder. The proof system is a slight modification of the system introduced in [BKO88]. However in that reference only a sub-language consisting of recursion free terms of ACP is considered.

The logical characterization gives us an interesting and strong proof technique for proving properties of the behavioural preorder. Like in [AH92] we use it to prove that the preorder is finitary in the sense of [Abr91]. However here we go a step further and prove the algebraicity, in the sense of [Hen88], of the preorder by using its modal characterization. This allows us to reduce the proof of the completeness of the proof system to only proving the completeness over finite or recursion free terms. This proof technique is interesting in itself as the algebraicity usually follows only as a corollary to a full abstractness result with respect to a denotational model defined in terms of an algebraic domain. See for instance [Hen88, AH92, HI93, Ing95] for this.

The structure of what remains of the paper is as follows: In the next section we define the readiness preorder as an abstraction on the stepwise semantics given by labelled transition systems. We also give a bisimulation like characterization of the preorder. Section 3 is devoted to a modal characterization of the preorder. In Section 4 we introduce the language $CCS_{\oplus}$ and apply the theory to it, introduce a proof system and prove its soundness and completeness with respect to the readiness preorder. We complete the paper by pointing out some directions for further work in Section 5.

2 Operational Semantics

We describe the behaviour of processes by means of a slight modification of the standard labelled transition system, *extended labelled transition system* [Hen88], where the internal actions are given by the transition relation $\rightarrowtail$ instead of the more standard notation $\xrightarrow{\tau}$.

Definition 2.1 We define an extended labelled transition system (LTS) as $\langle States, Lab, \longrightarrow, \rightarrowtail \rangle$ where

- $States$ is a set of states, ranged over by s,
- Lab is a set of labels, ranged over by l,
- $\longrightarrow \subseteq States \times Lab \times States$ is an external transition relation,
- $\rightarrowtail \subseteq States \times States$ is an internal transition relation.

We let Λ range over all LTS's. □

As usual, we write $(s, s') \in \rightarrowtail$ as $s \rightarrowtail s'$ (read "s may evolve spontaneously to s'") and $(s, l, s') \in \longrightarrow$ as $s \xrightarrow{l} s'$ (read "s may perform l and thereby become s'"). We define $\stackrel{\varepsilon}{\Longrightarrow}$ as $\rightarrowtail^*$, $\stackrel{l}{\Longrightarrow}$ (the weak l-action) as $\rightarrowtail^* \cdot \xrightarrow{l} \cdot \rightarrowtail^*$ and $\stackrel{\sigma}{\Longrightarrow}$ where $\sigma \in Lab^*$ similarly. We write $s \rightarrowtail$ if $s \rightarrowtail s'$ for some s' and $s \not\rightarrowtail$ if no such s' exists. In this case we say that s is a stable state. We say that $s \uparrow$ (read "s diverges" or "is divergent") if there is an infinite sequence of states $s_1, s_2, \ldots$ such that $s \rightarrowtail s_1 \rightarrowtail \ s_2 \ldots$. That $s \downarrow$ (read s "converges" or "is convergent") is defined as $\neg(s \uparrow)$. We also define $s \uparrow l$ (read "s diverges on l") as $(s \uparrow \vee \exists s'.s \stackrel{l}{\Longrightarrow} s' \wedge s' \uparrow)$ and $s \downarrow l$ as $\neg(s \uparrow l)$. We define

- $I(s) = \{l \in Lab | s \xrightarrow{l}\}$, the initial set of s
- $\mathcal{I}(S) = \{I(s) | s \in S\}$, the ready set of S,
- $I(S) = \bigcup \mathcal{I}(S)$, the initial set of S,
- $Int(s) = \{s' | s \rightarrowtail s'\}$,
- $Ext(s) = \{s' | \exists l \in Lab.s \xrightarrow{l} s'\}$,
- $Sort_w(s) = \{l \in Lab \,|\, \exists \sigma \in Act^*, s' \in States.\, s' \downarrow \wedge s \stackrel{\sigma}{\Longrightarrow} s' \xrightarrow{l}\}$.

We say that s is *finitely branching* if $Int(s) \cup Ext(s)$ is finite and that Λ is weakly finitely branching if all its convergent states are finitely branching. A state, s, is said to be *weakly sort finite* if $Sort_w(s)$ is finite. This definition extends to LTS's in the obvious way. If for all $s \in States$ the following holds:

$$s \xrightarrow{a} s' \text{ and } s \rightarrowtail s'' \text{ implies } s'' \xrightarrow{a} s'$$

we say that communication capabilities are preserved by $\rightarrowtail$. If this is the case then it is easy to see that if $s \downarrow$ and $s \xrightarrow{a} s'$ then there is a state $s'' \not\rightarrowtail$ such that $s \rightarrowtail^* s'' \xrightarrow{a} s'$. As explained in the Introduction we make the following assumption.

Assumption 2.2 *Throughout the paper we assume that all LTS's are weakly sort-finite and weakly finitely branching and that communication capabilities are preserved by* $\succ\!\!\rightarrow$.

In the Introduction we made the assumption that we are not able to test divergent processes, i.e. no tests terminate when applied to them, and that we only perform experiments on processes when they have reached a stable state. Therefore we model a convergent process as the set of stable states it can reach internally and a divergent process as the singleton set $\{\bot\}$. Furthermore, as we want to model when two processes s and t may pass the same tests, we have to compare the set of all possible stable states s may reach when performing a weak l-action to the set of all possible stable states t may reach when performing the same action. To model this we define the l-derative, $Der(l, S)$ of a set of stable states, S, where $l \in I(S)$, as the set of stable states which can be reached from the states in S by performing a weak l-action if all its states converge on l and $\{\bot\}$ otherwise. In this way any labelled transition system, Λ, induces a deterministic labelled transition system where the new states are either finite sets of stable states of the original one or the singleton set $\{\bot\}$. To ensure that this approach makes sense we prove the following property.

Lemma 2.3 *For all* $s \in States$ *the following holds.*

1. *If* $s \downarrow$ *then* $\{s' | s \succ\!\!\rightarrow^* s'\}$ *is finite.*
2. *If* $s \downarrow l$ *then* $\{s' | s \stackrel{l}{\Longrightarrow} s'\}$ *is finite.*

Proof Follows by a simple application of Köning's Lemma [Knu73]. □

Next we define some notation we use throughout the paper.

- $Stb(s) = \{s' | s \succ\!\!\rightarrow^* s' \not\succ\!\!\rightarrow\}$: the set of stable states of s,
- $Stb_w(s) = \begin{cases} Stb(s) & \text{if } s \downarrow \\ \{\bot\} & \text{if } s \uparrow \end{cases}$: the set of weakly stable states of s and
- $Stb = \{s | s \not\succ\!\!\rightarrow\}$: the set of stable states.

By Lemma 2.3, $Stb_w(s)$ is finite for all $s \in States$. We let $\mathcal{S} = \mathcal{P}^+_{fin}(Stb) \cup \{\{\bot\}\}$, where $\mathcal{P}^+_{fin}(Stb)$ denotes the family of finite non-empty subsets of Stb. For $S \in \mathcal{S}$ we define

- $S \uparrow$ iff $S = \{\bot\}$,
- $S \downarrow$ iff $\neg(S \uparrow)$,
- $S \uparrow l$ iff $S \uparrow$ or $\exists s \in S. s \uparrow l$,

- $S \downarrow l$ iff $\neg(S \uparrow l)$,
- for $l \in I(S)$ we let

$$Der(l,S) = \begin{cases} \{s' \mid \exists s \in S.\, s \stackrel{l}{\Longrightarrow} s' \\ and s' \not\longrightarrow\} & \text{if } S \downarrow l \\ \{\bot\} & \text{if } S \uparrow l \end{cases} : \text{ the } l\text{-derivative of } S.$$

Again Lemma 2.3 ensures that $Der : Lab \times \mathcal{S} \hookrightarrow \mathcal{S}$ is well-defined this way as a partial function. In what follows we will define the *readiness preorder*, $\sqsubseteq_{\mathcal{R}}$. For this purpose we give a formal definition of the class of tests we apply in the experiments. We also define predicates which tell when a process may or may not pass a given test. Because of our intuitionistic approach, i.e. that the divergent states are not testable, we need two predicates, one to record when a process may pass the test and another one to tell when it must fail.

Definition 2.4 (Readiness Tests) The set $\mathcal{T}$ of readiness tests is defined as the least set which satisfies:

1. $A \subseteq_{fin} Lab$ implies $A \in \mathcal{T}$,
2. $\gamma \in \mathcal{T}$, $l \in Lab$ implies $l.\gamma \in \mathcal{T}$.

The success functions $MayPass, MustFail : \mathcal{T} \times \mathcal{S} \longrightarrow Bool$ are defined as follows:

For all $S \in \mathcal{S}$

1. $MayPass(A,S) = S \downarrow \wedge A \in \mathcal{I}(S)$,
2. $MayPass(l.\gamma,S) = S \downarrow \wedge l \in I(S) \wedge MayPass(\gamma, Der(l,S))$.

1. $MustFail(A,S) = S \downarrow \wedge A \notin \mathcal{I}(S)$,
2. $MustFail(l.\gamma,S) = S \downarrow \wedge (l \in I(S) \Rightarrow MustFail(\gamma, Der(l,S)))$.

□

We define the readiness preorder by

Definition 2.5 (Readiness Preorder)

$$S \sqsubseteq_{\mathcal{R}} T \text{ iff } \forall \gamma \in \mathcal{T}.\ MayPass(\gamma,S) \Rightarrow MayPass(\gamma,T) \wedge MustFail(\gamma,S) \Rightarrow MustFail(\gamma,T).$$

and for $s,t \in State$

$$s \sqsubseteq_{\mathcal{R}} t \text{ iff } Stb_w(s) \sqsubseteq_{\mathcal{R}} Stb_w(t).$$

The derived equivalence is denoted by $=_{\mathcal{R}}$. □

Example 2.6 *Let $\gamma_1 = \{l\}$, $\gamma_2 = l.\emptyset$ and $S = \{l.\Omega\}$, where $\Omega \uparrow$. It is easy to see that $MayPass(\gamma_1, S)$ is true and $MustFail(\gamma_1, S)$ is not. On the other hand neither $MayPass(\gamma_2, S)$ nor $MustFail(\gamma_2, S)$ is true.*

We have the following result which is proved in Appendix A.

Theorem 2.7 *If the set of labels, Lab, is infinite then the equivalence $=_{\mathcal{R}}$ coincides with the readiness equivalence of [OH86].*

Next we give a bisimulation like characterization of the relation $\sqsubseteq_{\mathcal{R}}$, i.e. we obtain the preorder as the greatest fixed point to a monotonic endofunction over the complete lattice $(\mathcal{S} \times \mathcal{S}, \subseteq)$.

Definition 2.8 (Alternative Readiness Preorder) Let $\mathcal{F} : \mathcal{S} \times \mathcal{S} \longrightarrow \mathcal{S} \times \mathcal{S}$ be defined by:

$$\begin{aligned}\mathcal{F}(\mathcal{R}) = \{(S,T) |\, S \downarrow \text{ implies } & 1.\ T \downarrow, \\ & 2.\ \mathcal{I}(S) = \mathcal{I}(T) \text{ and} \\ & 4.\ \forall l \in I(S).(Der(l,S), Der(l,T)) \in \mathcal{R}\}.\end{aligned}$$

The alternative readiness preorder $\preceq$ is defined as the greatest fixed-point to $\mathcal{F}$. We define

$$s_1 \preceq s_2 \text{ iff } Stb_w(s_1) \preceq Stb_w(s_2).$$

□

$\mathcal{R}$ is said to be an alternative readiness relation if $\mathcal{R} \subseteq \mathcal{F}(\mathcal{R})$. By the standard fixed point theory [Tar55],

$$\preceq = \bigcup\{\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \,|\, \mathcal{R} \subseteq \mathcal{F}(\mathcal{R})\},$$

or equivalently

$$S_1 \preceq S_2 \text{ iff } (S_1, S_2) \in \mathcal{R} \text{ for some alternative readiness relation } \mathcal{R}.$$

The ω-alternative readiness preorder $\preceq_\omega$ is defined as follows.

1. $\preceq_0 = \mathcal{S} \times \mathcal{S}$,
2. $\preceq_{n+1} = \mathcal{F}(\preceq_n)$,
3. $\preceq_\omega = \bigcap_{n \in \omega} \preceq_n$

and as before

$$s_1 \preceq_\omega s_2 \text{ iff } Stb_w(s_1) \preceq_\omega Stb_w(s_2).$$

In what follows we state that the two preorders $\preceq$ and $\preceq_\omega$ coincide and that they actually are preorders.

Lemma 2.9

I. $\preceq = \preceq_\omega$.

II. $\preceq$ *is a preorder. We refer to its kernel as* $\simeq$.

Proof Obviously it is sufficient to prove the statements for $\preceq$ in $\mathcal{S} \times \mathcal{S}$ instead of $States \times States$. We prove each of the statements separately.

I. It is easy to see that $\preceq \subseteq \preceq_n$ for all n and therefore that $\preceq \subseteq \preceq_\omega$. To prove that $\preceq_\omega \subseteq \preceq$ it is sufficient to prove that $\preceq_\omega$ is a readiness preorder. This in turn follows easily from the fact that the transition relation is deterministic on $\mathcal{S}$.

II. To prove the reflexivity we note that $\preceq_n$ is reflexive for all n. To prove the transitivity it is sufficient to prove that $\preceq_n$ is transitive for all n. We proceed by induction on n. The base case $n = 0$ is trivial. For the inductive step we assume that $S_1 \preceq_{n+1} S_2$ and $S_2 \preceq_{n+1} S_3$, we will prove that $S_1 \preceq_{n+1} S_3$, i.e. that $S_1 \mathcal{F}(\preceq_n) S_3$. We proceed as follows.

Assume $S_1 \downarrow$.

1. This implies that $S_2 \downarrow$ and therefore that $S_3 \downarrow$.
2. Assume $S_1 \downarrow$, $S_2 \downarrow$ and $S_3 \downarrow$. As $S_1 \preceq_{n+1} S_2$, $\mathcal{I}(S_1) = \mathcal{I}(S_2)$. As $S_2 \preceq_{n+1} S_3$, $\mathcal{I}(S_2) = \mathcal{I}(S_3)$, i.e. $\mathcal{I}(S_1) = \mathcal{I}(S_3)$.
3. Again assume $S_1 \downarrow$, $S_2 \downarrow$ and $S_3 \downarrow$, $\mathcal{I}(S_1) = \mathcal{I}(S_2)$ and that $l \in I(S_1)$. Then $l \in I(S_2)$. Furthermore, by assumption, $Der(l, S_1) \preceq_n Der(l, S_2)$ and $Der(l, S_2) \preceq_n Der(l, S_3)$. By induction $Der(l, S_1) \preceq_n Der(l, S_3)$.

□

The following theorem states the equivalence between the two relations defined above.

Theorem 2.10 (Alternative characterization) *For all* $s, t \in States$

$$s \preceq t \text{ iff } s \sqsubseteq_{\mathcal{R}} t.$$

Proof It is sufficient to prove the theorem for $S, T \in \mathcal{S}$.

"only if": Assume that $S \preceq T$, we will prove that

$$\forall \gamma \in \mathcal{T}.\ 1.MayPass(\gamma, S) \Rightarrow MayPass(\gamma, T) \text{ and}$$
$$2.MustFail(\gamma, S) \Rightarrow MustFail(\gamma, T).$$

We proceed by induction on the definition of γ and consider each statement, 1. and 2., separately. First we note that $S \preceq T$ and $S \downarrow$ implies $I(T) = I(T)$. Now we proceed as follows.

1. For the base case assume $A \subseteq_{fin} Lab$. Then we have

$$\begin{aligned} MayPass(A,S) &\Rightarrow S\downarrow \wedge A \in \mathcal{I}(S) && \text{(by def. of } MayPass) \\ &\Rightarrow T\downarrow \wedge A \in \mathcal{I}(T) && \text{(as } S \preceq T) \\ &\Rightarrow MayPass(A,T) \quad . \end{aligned}$$

   For the inductive step assume $l \in Lab$ and $\gamma \in \mathcal{T}$.

$$\begin{aligned} MayPass(l.\gamma,S) &\Rightarrow S\downarrow \wedge l \in I(S) \wedge MayPass(\gamma, Der(l,S)) \\ &\qquad \text{(by def. of } MayPass) \\ &\Rightarrow T\downarrow \wedge l \in I(T) \wedge MayPass(\gamma, Der(l,T)) \\ &\qquad \text{(by ind. as } S \preceq T) \\ &\Rightarrow MayPass(l.\gamma,T). \end{aligned}$$

2. For the base case, as before, assume $A \subseteq_{fin} Lab$. We have

$$\begin{aligned} MustFail(A,S) &\Rightarrow S\downarrow \wedge A \notin \mathcal{I}(S) && \text{(by def. of } MustFail) \\ &\Rightarrow T\downarrow \wedge A \notin \mathcal{I}(S) && \text{(as } S \preceq T) \\ &\Rightarrow MustFail(A,T). \end{aligned}$$

   For the inductive step we have:

$$\begin{aligned} MustFail(l.\gamma,S) &\Rightarrow S\downarrow \wedge (l \in I(S) \Rightarrow MustFail(\gamma, Der(l,S)) \\ &\qquad \text{(by def. of } MustFail) \\ &\Rightarrow T\downarrow \wedge (l \in I(T) \Rightarrow MustFail(\gamma, Der(l,T)) \\ &\qquad \text{(by ind. as } S \preceq T \\ &\Rightarrow MustFail(l.\gamma,T). \end{aligned}$$

"if": It is sufficient to prove the following statement:

$$(\forall n. S \not\preceq_n T) \text{ implies } \exists \gamma \in \mathcal{T}.\ (MayPass(\gamma,S) \wedge \neg MayPass(\gamma,T)) \vee (MustFail(\gamma,S) \wedge \neg MustFail(\gamma,T)).$$

We prove this statement by induction on n where the base case $n = 0$ is trivial. For the inductive step we proceed by considering the following cases according to why $S \preceq_{n+1} T$ fails.

$S \downarrow$ but $T \uparrow$: Let $A \in \mathcal{I}(S)$. Then $MayPass(A, S)$ but $\neg MayPass(A, T)$.

$S \downarrow$ and $T \downarrow$ but $\mathcal{I}(S) \setminus \mathcal{I}(T) \neq \emptyset$: Let $A \in \mathcal{I}(S) \setminus \mathcal{I}(T)$. As before $MayPass(A, S)$ but $\neg MayPass(A, T)$.

$S \downarrow$ and $T \downarrow$ but $\mathcal{I}(T) \setminus \mathcal{I}(S) \neq \emptyset$: Let $A \in \mathcal{I}(T) \setminus \mathcal{I}(S)$. Then $MustFail(A, S)$ but $\neg MustFail(A, T)$.

$S \downarrow$, $T \downarrow$, $\mathcal{I}(S) = \mathcal{I}(T)$ and $l \in I(S) = I(T)$ but $Der(l, S) \not\preceq_n Der(l, T)$: By induction there is a $\gamma \in \mathcal{T}$ such that either of the following holds:

$MayPass(\gamma, Der(l, S))$ but $\neg MayPass(\gamma, Der(l, T))$:
In this case $MayPass(l.\gamma, S)$ but $\neg MayPass(l.\gamma, T)$.

$MustFail(\gamma, Der(l, S))$ but $\neg MustFail(\gamma, Der(l, T))$:
Then $MustFail(l.\gamma, S)$ but $\neg MustFail(l.\gamma, T)$.

□

As the preorders $\preceq$ and $\sqsubseteq_{\mathcal{R}}$ coincide from now on we refer to both of them as the readiness preorder.

3 Modal Characterization

In this section we give a modal characterization of the readiness preorder. This characterization will be an important tool in proving properties of the preorder. The modal logic $\mathcal{L}$ is generated by the following syntax

$$\phi ::= \mathcal{A} \mid [l]\phi \mid \phi \wedge \phi.$$

where $\mathcal{A} \subseteq \mathcal{P}_{fin}(Lab)$[1]. The satisfaction relation $\models \subseteq \mathcal{S} \times \mathcal{L}$, is defined inductively by:

1. $S \models \mathcal{A}$ iff $S \downarrow$ and $\mathcal{I}(S) = \mathcal{A}$,
2. $S \models [l]\phi$ iff $S \downarrow \wedge (l \in I(S) \Rightarrow Der(l, S) \models \phi)$,
3. $S \models \phi_1 \wedge \phi_2$ iff $S \models \phi_1$ and $S \models \phi_2$.

As $\wedge$ is commutative and associative we often write $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n$ without giving the bracketing. Furthermore we let

$$s \models \phi \text{ iff } Stb_w(s) \models \phi.$$

In this case we say that s satisfies ϕ. Note that a state or a set of states satisfies a formula in $\mathcal{L}$ only if it is convergent. This implies that $S \models [l]\phi$ and $l \in I(S)$ only if $S \downarrow l$. The modal depth, $md(\phi)$, of a formula, ϕ is defined by

[1] the family of finite subsets of Lab

- $md(\mathcal{A}) = 1$,
- $md([l]\phi) = 1 + md(\phi)$ for $l \in Lab$,
- $md(\phi_1 \wedge \phi_2) = max\{md(\phi_1), md(\phi_2)\}$.

Also we define $\mathcal{L}_n = \{\phi \in \mathcal{L} | md(\phi) \leq n\}$ for $n \geq 0$. We let $\mathcal{L}(S) = \{\phi \in \mathcal{L} | S \models \phi\}$ and $\mathcal{L}_n(S) = \mathcal{L}_n \cap \mathcal{L}(S)$. We define $\mathcal{L}(s)$ and $\mathcal{L}_n(s)$ in the same way. The modal characterization of the readiness preorder is the content of the following theorem.

Theorem 3.1

1. *For all S, T, $\forall n.\, S \preceq_n T$ iff $\mathcal{L}_n(S) \subseteq \mathcal{L}_n(T)$.*
2. *For all s, t, $s \preceq t$ iff $\mathcal{L}(s) \subseteq \mathcal{L}(t)$*

Proof To prove 1. we proceed as follows:

"only if": We prove the statement by induction on n. The base case $n = 0$ is obvious. For the inductive step assume $S \preceq_{n+1} T$ and $S \models \phi$, where $md(\phi) = n + 1$. We will prove that $T \models \phi$. We proceed by structural induction on ϕ.

$\phi \equiv \mathcal{A}$:

$$\begin{array}{lll} S \models \mathcal{A} & \Rightarrow S \downarrow \wedge \mathcal{I}(S) = \mathcal{A} & \text{(Pr. def. of } \models) \\ & \Rightarrow T \downarrow \wedge \mathcal{I}(T) = \mathcal{A} & (\text{ as } S \preceq_{n+1} T) \\ & \Rightarrow T \models \mathcal{A} & \text{(Pr. def. of } \models) \end{array}$$

$\phi \equiv [l]\psi$: We recall that $S \models [l]\psi$ is equivalent to

$$S \downarrow \wedge (l \in I(S) \Rightarrow Der(l, S) \models \psi). \tag{1}$$

We will prove that (1) holds for S replaced by T. As $S \preceq_{n+1} T$, $T \downarrow$. So assume $l \in I(T)$. Then $l \in I(S)$ and by (1), $Der(l, S) \models \psi$. As $Der(l, S) \preceq_n Der(l, T)$ and $md(\psi) = n$, by the outer induction $Der(l, T) \models \psi$. This implies $T \models [l]\psi$.

$\phi \equiv \phi_1 \wedge \phi_2$: $S \models \phi_1 \wedge \phi_2$ implies $S \models \phi_1$ and $S \models \phi_2$. By the structural induction $T \models \phi_1$ and $T \models \phi_2$, i.e. $T \models \phi_1 \wedge \phi_2$.

"if": It is sufficient to prove that for all n.

$$S \not\preceq_n T \Rightarrow \exists \phi \in \mathcal{L}_n.\, S \models \phi \wedge T \not\models \phi.$$

We prove this statement by induction on n. The base case for the induction is vacuously true. For the inductive step it is sufficient to consider following cases according to why $S \not\preceq_{n+1} T$.

$S\downarrow$ but $T\uparrow$: Then $S \models \mathcal{I}(S)$ but $T \not\models \mathcal{I}(S)$

$S\downarrow$ and $T\downarrow$ and $\mathcal{I}(S) \neq \mathcal{I}(T)$: Again $S \models \mathcal{I}(S)$ but $T \not\models \mathcal{I}(S)$.

$S\downarrow$, $T\downarrow$, $\mathcal{I}(S) = \mathcal{I}(T)$ and $l \in I(T) = I(S)$ but $Der(l,S) \not\preceq_n Der(l,T)$: By induction there is a $\psi \in \mathcal{L}_n$ such that $Der(l,S) \models \psi$ but $Der(l,T) \not\models \psi$. This implies that $S \models [l]\psi$ but $T \not\models [l]\psi$.

Statement 2. follows directly from statement 1., Lemma 2.9 and the definitions of $s \preceq t$ and $\mathcal{L}(s)$. □

Theorem 3.1 says that the preorder is fully characterized by the logic $\mathcal{L}$ interpreted over $\models$. Furthermore from the proof we see that the operator $\wedge$ is not needed in the characterization. However in what follows we show that each $S \in \mathcal{S}$ that corresponds to a finite convergent process may be characterized up to $\preceq$ by one single formula in $\mathcal{L}$, i. e. every such S has a *characteristic formula* in the sense of [SI94]. To obtain this result we need the $\wedge$ operator. We start by defining $dpth(S) = sup\{|s| \,|S \stackrel{\sigma}{\Longrightarrow}\}$. We say that S is of *finite depth* if $dpth(S) < \infty$.

Definition 3.2 (Characteristic formula) Let $D \in \mathcal{S}$ be convergent and of finite depth. We define the characteristic formula, ϕ_D, for D by induction on $dpth(D)$ by:

$dpth(D) = 0$: $\phi_D \equiv \{\emptyset\}$.

$dpth(D) = n+1$: $\phi_D \equiv \mathcal{I}(D) \wedge \bigwedge\{[l]\phi_{Der(l,D)} | l \in I(D) \wedge D\downarrow l\}$.

□

We have the following characterization theorem

Theorem 3.3 *For every convergent $D \in \mathcal{S}$ of finite depth following holds:*

1. $D \models \phi_D$ *and*
2. $\forall S.\, S \models \phi_D \Rightarrow D \preceq S$.

Proof

1. We prove the statement by induction on $dpth(D)$.

   $dpth(D) = 0$: Obviously $D \models \{\emptyset\}$.

   $dpth(D) = n+1$: Again it is obvious that $D \models \mathcal{I}(D)$. Furthermore, if $l \in I(D)$ and $D\downarrow l$, then $Der(l,D)\downarrow$. By induction $Der(l,D) \models \phi_{Der(l,D)}$ which in turn implies that $D \models [l]\phi_{Der(l,S)}$. We have therefore that $D \models \bigwedge\{[l]\phi_{Der(l,D)} | l \in I(D) \wedge D\downarrow l\}$. This completes the proof of $D \models \phi_D$.

2. Next assume that $S \models \phi_D$. We will prove that $D \preceq S$ by showing that the defining clauses for $\preceq$ are satisfied. Again we prove the statement by induction on $dpth(D)$.

   $dpth(D) = 0$: It is easy to check that $S \models \{\emptyset\}$ implies $D \preceq S$.

   $dpth(D) = n + 1$: Now we proceed as follows: We know that $D \downarrow$.

   (a) As $S \models \phi_D$, $S \downarrow$.

   (b) As $S \models \mathcal{I}(D)$, $\mathcal{I}(S) = \mathcal{I}(D)$

   (c) Let $l \in I(S) = I(T)$. We have the following possibilities:

   - $Der(l, D) \uparrow$: Then $Der(l, D) \preceq Der(l, S)$.
   - $Der(l, D) \downarrow$: As $S \models \bigwedge\{[l]\phi_{Der(l,D)} | l \in I(D) \wedge D \downarrow l\}$ this implies $Der(l, S) \models \phi_{Der(l,D)}$. By induction we get that $Der(l, D) \preceq Der(l, S)$.

□

4 Application to Regular CCS

In this section we give the syntax for a sublanguage of a slight modification of the standard CCS and an operational semantics based on the ideas described in the previous section.

4.1 Syntax

The language we investigate in this study is the set of regular processes of the language $CCS_{\oplus}$, i. e. CCS where τ is replaced by the internal choice operator $\oplus$. This language is studied in more detail for testing based semantics in [Hen88].

In the definition of the syntax we assume a predefined countably infinite set of process variables $PVar$ ranged over by x,y, etc. and a set of of actions, Act, ranged over by a,b, etc. The set of allowed operators is Θ, Ω of arity 0, $a.\_$, $a \in Act$ of arity 1 and $\oplus$ and $+$ of arity 2. We use Σ to denote this collection of operators and Σ_k those of arity k. The set of (process) terms is then defined by the BNF-definition

$$t \quad ::= \quad \Theta \mid \Omega \mid t + t \mid t \oplus t \mid a.t \mid x \mid \mathtt{rec}x.t.$$

The construction $\mathtt{rec}x.\_$ binds occurrences of process names which gives rise in the usual way to free and bound names and to closed and open terms. We use $t[u/x]$ to denote the term which results from substituting the term u for every free occurrence of x in t. We will sometimes use a more general form of substitution. If ρ is a mapping from $PVar$ to the set of terms then $t\rho$ is the term which results

1. $a.p \xrightarrow{a} p$

2. $p \xrightarrow{a} p'$ implies $\quad p+q \xrightarrow{a} p'$, $\; q+p \xrightarrow{a} p'$

3. $p \oplus q \rightarrowtail p$, $\; p \oplus q \rightarrowtail q$

4. $\Omega \rightarrowtail \Omega$

5. $\mathtt{rec}x.t \rightarrowtail t[\mathtt{rec}x.t/x]$

6. $p \rightarrowtail p'$ implies $\quad p+q \rightarrowtail p'+q$, $\; q+p' \rightarrowtail q+p'$

Figure 1: Rules for $\rightarrowtail$ and $\xrightarrow{a}$

from simultaneously substituting $\rho(x)$ for each free occurrence of x in t. The set of processes, i.e. closed terms, is denoted by $Proc$ ranged over by p and that of recursion free processes by $FinProc$, ranged over by d.

4.2 The Operational Semantics

The concrete operational semantics for $Proc$ is given by Figure 1. It is easy to check that the labelled transition system defined this way, $\langle Proc, Act, \longrightarrow, \rightarrowtail \rangle$, satisfies the conditions described in Assumption 2.2. It turns out that the readiness preorder is a precongruence with respect to the operators in Σ.

Lemma 4.1 *$\preceq$ is a precongruence with respect to the operators in Σ.*

Proof To prove that $\preceq$ is a precongruence it is sufficient to prove that for all n, $\preceq_n$ is preserved by the operators. We proceed by induction on n where the base case, $n = 0$, is trivial. For the inductive step we consider each of the operators separately:

$a.\_$: First we note that for any p, $Stb_w(a.p) = \{a.p\}$. Now it is straight forward to show that $Stb_w(p) \preceq_{n+1} Stb_w(q)$ implies $\{a.p\} \preceq_{n+1} \{a.q\}$.

$\oplus$: Here we note that for any p_1 and p_2, $p_1 \oplus p_2 \downarrow$ iff $p_1 \downarrow$ and $p_2 \downarrow$. Furthermore, if $p_1 \oplus p_2 \downarrow$, then $Stb_w(p_1 \oplus p_2) = Stb_w(p_1) \cup Stb_w(p_2)$. Then we note that

$$\forall S, T, U \neq \{\bot\}.\, S \preceq T \Rightarrow S \cup U \preceq T \cup U. \tag{2}$$

Now the result follows easily by induction using these observations.

$+$: As before, for any p_1 and p_2, $p_1 + p_2 \downarrow$ iff $p_1 \downarrow$ and $p_2 \downarrow$. Now we proceed as follows:

Assume $Stb_w(p_1) \preceq_{n+1} Stb_w(q_1)$ and $Stb_w(p_2) \preceq_{n+1} Stb_w(q_2)$. We will prove that $Stb_w(p_1 + p_2) \preceq_{n+1} Stb_w(q_1 + q_2)$. So assume $Stb_w(p_1 + p_2) \downarrow$.

1. Then $Stb_w(p_1) \downarrow$ and $Stb_w(p_2) \downarrow$. This implies that $Stb_w(q_1) \downarrow$ and $Stb_w(q_2) \downarrow$ which in turn implies that $Stb_w(q_1 + q_2) \downarrow$.
2. Next assume that $Stb_w(p_1 + p_2) \downarrow$ and $Stb_w(q_1 + q_2) \downarrow$.
   (a) First let $p \in Stb(p_1 + p_2)$. This implies that $p = p_1' + p_2'$ where $p_1' \in Stb(p_1)$ and $p_2' \in Stb(p_2)$. Then there is a $q_1' \in Stb(q_1)$ such that $I(p_1') = I(q_1')$ and $q_2' \in Stb(q_2)$ such that $I(p_2') = I(q_2')$. This implies that $I(p) = I(p_1' + p_2') = I(q_1' + q_2') = I(q)$ where $q_1' + q_2' \in Stb(q_1 + q_2)$.
   (b) Now let $Stb_w(p_1 + p_2) \downarrow$, $Stb_w(q_1 + q_2) \downarrow$ and $q \in Stb_w(q)$. As before we may conclude that $I(p) = I(q)$ for some $p \in Stb_w(p_1 + p_2)$.
   (a) and (b) imply that $\mathcal{I}(Stb(p_1 + p_2)) = \mathcal{I}(Stb(q_1 + q_2))$.
3. Finally assume that $Stb_w(p_1 + p_2) \downarrow$, $Stb_w(q_1 + q_2) \downarrow$ and $a \in I(Stb(p_1 + p_2)) = I(Stb(q_1 + q_2))$.
   (a) First assume that $p_1 + p_2 \uparrow a$. Then

$$Der(a, Stb_w(p_1 + p_2)) = \{\bot\}$$

   and therefore

$$Der(a, Stb_w(p_1 + p_2)) \preceq Der(a, Stb_w(q_1 + q_2)).$$

   (b) Next assume that $p_1 + p_2 \downarrow a$. If $a \notin I(p_i)$ for either $i = 1$ or $i = 2$, say $i = 1$ then $a \notin I(q_1)$ and

$$Der(a, Stb(p)) = Der(a, Stb(p_2)) \preceq Der(a, Stb(q_2)) = Der(a, Stb(q)).$$

   So assume $a \in I(p_1) \cap I(p_2)$. Then

$$Der(a, Stb(p_1 + p_2)) = Der(a, Stb(p_1)) \cup Der(a, Stb(p_2)).$$

As $Stb_w(p_1) \preceq_{n+1} Stb_w(q_1)$ and $Stb_w(p_2) \preceq_{n+1} Stb_w(q_2)$, by (2)

$$Der(a, Stb_w(p_1 + p_2)) =$$
$$Der(a, Stb_w(p_1)) \cup Der(a, Stb_w(p_2)) \preceq_n$$
$$Der(a, Stb_w(q_1)) \cup Der(a, Stb_w(q_2)) =$$
$$Der(a, Stb_w(q_1 + q_2)).$$

□

4.3 The Proof System

In this section we introduce a proof system to reason about process behaviour. The proof system consists of a set of equations, Figure 2, and an inference system, Figure 3. The equations are a slight modification of those introduced in [BKO88] due to a different language as in that reference a subset of recursive free processes of ACP is considered. Furthermore the proof system is almost the same as for the must testing preorder [Hen88] with the following differences.

- The (in)equations

  $$(X + Y) \oplus Z = (X \oplus Z) + (Y \oplus Z)$$

  and

  $$X \oplus Y \sqsubseteq X$$

  are omitted as they are not sound with respect to the readiness preorder.

- The equation

  $$(a.X + Y) \oplus (a.Z + W) = (a.X + Y) \oplus (a.X + W) \oplus (a.Z + W)$$

  has been added here but is derivable in the proof system that characterizes the must testing.

For the motivation of these equations and the inference rules see the references above.

The syntactical approximations p^n that occur in the (ω)-rule are taken directly from [Hen88] and are defined as follows:

- $p^0 = \Omega$
- p^{n+1} is defined inductively by:
  - $(op(\underline{p}))^{n+1} = op(\underline{p}^{n+1})$ for $op \in \Sigma$,

– $(\mathsf{rec}x.u)^{n+1} = u^{n+1}[(\mathsf{rec}x.u)^n/x]$.

We refer to the set of equations as E and let $\sqsubseteq_E$ denote the preorder derived from E and the finitary inference rules (*least*)–(*axiom*) in Figure 3, while $\sqsubseteq_{E^{-\omega}_{rec}}$ is obtained by adding the rule (*fix*) and $\sqsubseteq_{E_{rec}}$ is obtained by further adding the rule (ω), i.e. the full system. The discussion above shows that these preorders are strictly stronger than the corresponding ones for the must testing.

The following partial soundness result follows as an easy consequence of the definition of $\preceq$.

Lemma 4.2 (Partial soundness) *The proof system consisting of the equations and the inference rules (least)–(fix) is sound with respect to the stable state preorder, i.e.*

$$\forall p, q.\, p \sqsubseteq_{E^{-\omega}_{rec}} q \text{ implies } p \preceq q.$$

Proof The inference rules (*least*)–(*congr*) only state that the relation is as precongruence with Ω as a least element and are therefore sound for $\preceq$. Also the rule (*fix*) is obviously sound for $\preceq$. Therefore it only remains to proof the soundness of the rule (*axiom*), i.e. to check whether the equations are sound with respect to $\preceq$. This in turn is straight forward and is left to the reader. □

Here we would like to point out that, at this point, we have not stated the soundness of the ω-rule as the proof for this is non trivial and will be dealt with later. In what remains of the paper we use the equations ($\oplus$1)–($\oplus$3) and (+1)–(+4) to rewrite process terms without further explanation. Now we will state a standard result that holds for all proof systems of this kind. For justification see for instance [Hen88].

Lemma 4.3 *For all d, p,*

1. $\forall n.p^n \sqsubseteq_{E^{-\omega}_{rec}} p$,
2. $d \sqsubseteq_{E_{rec}} p$ *implies* $d \sqsubseteq_{E^{-\omega}_{rec}} p$,
3. $d \sqsubseteq_{E_{rec}} p$ *implies* $\exists n.d \sqsubseteq_E p^n$.

Proof All the statements are standard and a justification for them may be found in for instance in [Hen88]. □

Lemma 4.4 *The following equation is derivable from E.*

$$(a.X + Y) \oplus (a.Z + W) = (a.(X \oplus Z) + Y) \oplus (a.(X \oplus Z) + W) \qquad (Der).$$

$\oplus 1$	$X \oplus (Y \oplus Z) = (X \oplus Y) \oplus Z$
$\oplus 2$	$X \oplus Y = Y \oplus X$
$\oplus 3$	$X \oplus X = X$
$\oplus \Omega$	$X \oplus \Omega = \Omega$
$+1$	$X + (Y + Z) = (X + Y) + Z$
$+2$	$X + Y = Y + X$
$+3$	$X + X = X$
$+4$	$X + \Theta = X$
$+\Omega$	$X + \Omega = \Omega$
$pre + \oplus 1$	$a.X + a.Y = a.(X \oplus Y)$
$pre + \oplus 2$	$a.X \oplus a.Y = a.(X \oplus Y)$
$+ \oplus 1$	$X + (Y \oplus Z) = (X + Y) \oplus (X + Z)$
$+ \oplus 2$	$(a.X + Y) \oplus (a.Z + W) = (a.X + Y) \oplus (a.X + W) \oplus (a.Z + W)$

Figure 2: Equations

Proof The equation may be derived as follows:

$$
\begin{array}{ll}
(a.X + Y) \oplus (a.Z + W) = & \\
(a.X + Y) \oplus (a.X + W) \oplus (a.Z + Y) \oplus (a.Z + W) = & (+ \oplus 2) \\
(a.X + (Y \oplus W)) \oplus (a.Z + (Y \oplus W)) = & (+ \oplus 1) \\
(a.X \oplus a.Z) + (Y \oplus W) = & (+ \oplus 1) \\
a.(X \oplus Z) + (Y \oplus W) = & (pre + \oplus 2) \\
(a.(X \oplus Z) + Y) \oplus (a.(X \oplus Z) + W). & (+ \oplus 1)
\end{array}
$$

□

4.4 Finitariness and Algebraicity

In this subsection we will show that the preorder $\preceq$ is finitary in the sense of Abramsky [Abr91]. The proof is very similar to a proof of the same property for a bisimulation based preorder in [AH92] and like in that reference we use the logical characterization. However we go a step further and show that the preorder is algebraic in the sense of [Hen88] (basically that the (ω)-rule is sound) using similar techniques and by using the characteristic formula. This proof is of

$$(\textit{least}) \quad \Omega \sqsubseteq X$$

$$(\textit{preord}) \quad t \sqsubseteq t \qquad\qquad \frac{t \sqsubseteq u,\ u \sqsubseteq v}{t \sqsubseteq v}$$

$$(\textit{congr}) \quad \frac{t_i \sqsubseteq u_i}{op(\underline{t}) \sqsubseteq op(\underline{u})} \qquad \text{for every } op \in \sum$$

$$(\textit{axiom}) \quad \frac{}{t\rho \sqsubseteq u\rho} \qquad \text{for every equation } t \sqsubseteq u \text{ and closed substitution } \rho$$

$$(\textit{fixp}) \quad \frac{}{recx.t = t[recx.t/x]}$$

$$(\omega) \quad \frac{\forall n.\, t^n \sqsubseteq u}{t \sqsubseteq u}$$

Figure 3: Proof System

some theoretical interest in itself as the algebraicity of the behavioural preorder, and the soundness of the ω-rule are proved by using properties of the modal logics, the operational semantics and of the proof system where the rule (ω) is omitted. Normally this result follows as a consequence of a full abstractness of the behavioural preorder with respect to a denotational model in terms of an algebraic cpo [Hen88, AH92, HI93, Ing95]. Proving the algebraicity directly allows us to reduce the proof of completeness of the proof system over the full language to a completeness proof only over recursion free processes. We start by giving Abramsky's definition of when a preorder is finitary.

Definition 4.5 (Finitariness) The finitary part, $\leq^F$, of a preorder, $\leq$, is defined by

$$p \leq^F q \text{ iff } (\forall d.d \leq p \Rightarrow d \leq q).$$

A preorder $\leq$ is finitary if $\leq^F = \leq$. □

We need the following lemma:

Lemma 4.6

1. *For all $p \in Proc$, $p \not\longmapsto$ implies*

$$p =_{E_{rec}^{-\omega}} \sum\{a.p' | p \xrightarrow{a} p'\}.$$

2. *For all $p \in Proc$, $p \downarrow$ implies*

$$p =_{E_{rec}^{-\omega}} \sum\!\!\!\!\!\!\bigcirc\, Stb(p).$$

3. *For all $p \in Proc$, $p \not\longmapsto$ and $p \downarrow a$ implies*

$$p =_{E_{rec}^{-\omega}} \sum\{a.\sum\!\!\!\!\!\!\bigcirc\, Der(a, \{p\}) | a \in I(p)\}.$$

Proof

1. We will prove the statement by structural induction on p. The cases $p \equiv \Omega, p_1 \oplus p_2, recx.u$ are trivial as p is not stable. Also the cases $p \equiv \Theta, a.p_1$ follow immediately. For the only remaining case, $p \equiv p_1 + p_2$, we proceed as follows.

   First we note that $p \xrightarrow{a} p_1'$ iff $p_1 \xrightarrow{a} p_1'$ or $p_2 \xrightarrow{a} p_1'$. By induction we have

$$\begin{array}{l} p \equiv p_1 + p_2 =_{E_{rec}^{-\omega}} \\ \sum\{a.p_1' | p_1 \xrightarrow{a} p'\} + \sum\{a.p_2' | p_2 \xrightarrow{a} p'\} =_{E_{rec}^{-\omega}} \\ \sum\{a.p' | p \xrightarrow{a} p'\}. \end{array}$$

2. First we note that the statement is true if $p \not\longmapsto$. Then we prove by structural induction that

$$p \downarrow \wedge p \rightarrowtail \text{ implies } p =_{E_{rec}^{-\omega}} \sum\!\!\!\!\!\!\bigcirc\, \{p' | p \rightarrowtail p'\}.$$

   For the only nontrivial case, $p \equiv p_1 + p_2$, we proceed as follows.

   Assume $p_1 + p_2 \rightarrowtail p'$. Then either $p' \equiv p_1' + p_2$ where $p_1 \rightarrowtail p_1'$ or $p' \equiv p_1 + p_2'$ where $p_2 \rightarrowtail p_2'$. By the equation $(\oplus 3)$, structural induction and the equation $(+ \oplus 1)$, we have

$$\begin{array}{ll} p_1 + p_2 & =_{E_{rec}^{-\omega}} (p_1 + p_2) \oplus (p_1 + p_2) \\ & =_{E_{rec}^{-\omega}} (\sum\!\!\!\!\!\!\bigcirc\, \{p_1' | p_1 \rightarrowtail p_1'\} + p_2) \oplus (p_1 + \sum\!\!\!\!\!\!\bigcirc\, \{p_2' | p_2 \rightarrowtail p_2'\}) \\ & =_{E_{rec}^{-\omega}} \sum\!\!\!\!\!\!\bigcirc\, \{p_1' + p_2 | p_1 \rightarrowtail p_1'\} \oplus \sum\!\!\!\!\!\!\bigcirc\, \{p_1 + p_2' | p_2 \rightarrowtail p_2'\} \\ & =_{E_{rec}^{-\omega}} \sum\!\!\!\!\!\!\bigcirc\, \{p' | p_1 + p_2 \rightarrowtail p'\}. \end{array}$$

   Now the result follows easily by induction on the internal depth of p, i.e. $max\{n | \exists p'.p \rightarrowtail^n p'\}$.

3. Let $der(a,p) = \{p' | p \xrightarrow{a} p'\}$. By 1. and equation $(pre + \oplus 1)$,

$$p =_{E^{-\omega}_{rec}} \sum\{a.\sum der(a,p) | a \in I(p)\}.$$

Furthermore

$$Der(a,\{p\}) = Stab(\sum der(a,p)).$$

By assumption, $\sum der(a,p) \downarrow$ and thus, by 2.

$$\sum der(a,p) = \sum Der(a,\{p\})$$

and the result follows.

□

The key to the proof of the finitariness of $\preceq$ is the following proposition which also is proved in [AH92] where, as mentioned before, a CCS-like language and prebisimulation are considered.

Proposition 4.7 *For all p and ϕ, if $p \models \phi$ then there is a finite d such that $d \sqsubseteq_{E^{-\omega}_{rec}} p$ and $d \models \phi$.*

Proof We prove the statement by structural induction on ϕ and we proceed as follows:

$\phi \equiv \mathcal{A}$: $p \models \mathcal{A}$ means that $p \downarrow$ and $\mathcal{I}(Stb(p)) = \mathcal{A}$. Let

$$d \equiv \sum\{\sum\{a.\Omega | a \in I(p')\} | p' \in Stb(p)\}.$$

Obviously $d \models \mathcal{A}$. Furthermore

$$\begin{aligned} d &\equiv \sum\{\sum\{a.\Omega | a \in I(p')\} | p' \in Stb(p)\} \\ &\sqsubseteq_{E^{-\omega}_{rec}} \sum\{\sum\{a.p'' | p' \xrightarrow{a} p''\} | p' \in Stb(p)\} \\ &\sqsubseteq_{E^{-\omega}_{rec}} p \qquad \text{(by L.4.6.1,2)} \end{aligned}$$

$\phi \equiv [a]\psi$: First we note that $p \models [a]\psi$ implies $p \downarrow$. If $a \notin I(Stb(p))$ then d defined as in the previous case satisfies the conditions. So assume $a \in I(Stb(p))$. Then $Der(a, Stb(p)) \models \psi$ and thus $p' \equiv \sum Der(a, Stb(p)) \models \psi$. By induction there is a d' such that $d' \models \psi$ and $d' \sqsubseteq_{E^{-\omega}_{rec}} p'$. Now let

$$d \equiv \sum\{\sum\{b.d_b | b \in I(p')\} | p' \in Stb(p)\},$$

where $d_a \equiv d'$ and $d_b \equiv \Omega$ if $b \neq a$. It is easy to see that $d \models [a]\psi$. Furthermore

$$Der(a, Stb(p)) = \bigcup\{Der(a,\{p'\}) | p' \in Stb(p)\}.$$

Now we have

$$\begin{aligned} d &\equiv \sum\{\sum\{b.d_b | b \in I(p')\} | p' \in Stb(p)\} \\ &\sqsubseteq_{E^{-\omega}_{rec}} \sum\{\sum b.\{\sum Der(b, Stb(p)) | b \in I(p')\} | p' \in Stb(p)\} \text{ (by } (least)) \\ &=_{E^{-\omega}_{rec}} \sum\{\sum\{b.\sum Der(b, \{p'\}) | b \in I(p')\} | p' \in Stb(p)\} \quad \text{(by } (Der)) \\ &=_{E^{-\omega}_{rec}} \sum\{p' | p' \in Stb(p)\} \qquad \text{(by Lem. 4.6.3)} \\ &=_{E^{-\omega}_{rec}} p \qquad \text{(by Lem. 4.6.1).} \end{aligned}$$

$\phi \equiv \phi_1 \wedge \phi_2$: We recall that $p \models \phi_1 \wedge \phi_2$ iff $p \models \phi_1$ and $p \models \phi_2$. By the structural induction there are d_1 and d_2 such that $d_i \models \phi_i$ and $d_i \sqsubseteq_{E^{-\omega}_{rec}} p$ for $i = 1, 2$. By Lemma 4.3, for $i = 1, 2$ there is an n_i such that $d_i \sqsubseteq_E p^{n_i}$. Now let $n = max\{n_1, n_2\}$ and $d \equiv p^n$. Then $d_i \sqsubseteq_{E^{-\omega}_{rec}} d \sqsubseteq_{E^{-\omega}_{rec}} p$. As $d_i \models \phi_i$, this and Theorem 3.1 imply that $d \models \phi_1 \wedge \phi_2$.

□

Now we get that $\preceq$ is finitary and algebraic as a corollary to Proposition 4.7. The finitariness is proved basically in the same way as a similar result in [AH92].

Corollary 4.8 (Finitariness) $\preceq^F = \preceq = \preceq_\omega$.

Proof It is sufficient to prove that $\preceq^F = \preceq$ as the second equality is the content of Lemma 2.9. Obviously $\preceq \subseteq \preceq^F$. To prove the opposite inclusion, by Theorem 3.1 it is sufficient to prove that

$$p \preceq^F q \Rightarrow \mathcal{L}(p) \subseteq \mathcal{L}(q).$$

To prove this we proceed as follows.

Assume that $(\forall d \downarrow . d \preceq p \Rightarrow d \preceq q)$ and that $p \models \phi$. We will prove that $q \models \phi$. By Proposition 4.7 there is a d such that $d \sqsubseteq_{E^{-\omega}_{rec}} p$ and $d \models \phi$. By the partial soundness $d \preceq p$, by our assumption $d \preceq q$ and by Theorem 3.1, $q \models \phi$.

□

In the proof of the algebraicity we take advantage of the characteristic formulae for processes of finite depth, that was defined for sets of states. We extend this definition to recursion free processes by

$$\forall d \downarrow . \phi_d \equiv \phi_{Stb(d)}.$$

It is easy to see that ϕ_d is a characteristic formula for d in the sense of Theorem 3.3. The algebraicity of $\preceq$ is the content of the following corollary.

Corollary 4.9 (Algebraicity)

$$\forall p, d.\, d \preceq p \Rightarrow \exists n. d \preceq p^n.$$

Proof If $d \uparrow$ we are done. So assume that $d \downarrow$ and $d \preceq p$. Let ϕ_d be the characteristic formula for d. Then $d \models \phi_d$ and therefore $p \models \phi_d$, by Theorem 3.1. By Proposition 4.7, there is a recursion free d' such that $d' \sqsubseteq_{E^{-\omega}_{rec}} p$ and $d' \models \phi_d$. Therefore $d \preceq d' \sqsubseteq_{E^{-\omega}_{rec}} p$. Now, by Lemma 4.3, there exists an n such that $d' \sqsubseteq_E p^n$. This implies $d \preceq p^n$. □

From these corollaries we may derive the soundness of the (ω)-rule.

Corollary 4.10 (Soundness of the ω-rule) $(\forall n. p^n \preceq q)$ *implies* $p \preceq q$.

Proof By Corollary 4.8 it is sufficient to prove that

$$(\forall n. p^n \preceq q) \text{ implies } (\forall d. d \preceq p \Rightarrow d \preceq q).$$

So assume $(\forall n. p^n \preceq q)$ and $d \preceq p$. By Corollary 4.9, $d \preceq p^n$ for some n and by our assumption we get $d \preceq q$. □

4.5 Completeness of the Proof System

In this final subsection we will prove the completeness of the proof system with respect to the readiness preorder, $\preceq$. By the algebraicity of the preorder, stated in Corollary 4.9, the proof of the completeness for the proof system with respect to it may be reduced to proving the completeness for finite processes and then applying the inference rule (ω). The proof of the completeness for the finite processes is, as usual, based on the notion of normal forms.

Definition 4.11 (Normal forms) An $n \in FinProc$ is in a normal form if either $n \equiv \Omega$, $n \equiv \Theta$ or it is of the form $n \equiv \sum_{j \leq N} \sum_{i \leq N_j} a^j_i.n^j_i$ where the following holds for all $i, i_1, i_2, j, k, l, :$

1. n^j_i is a normal form,
2. if $i_1 \neq i_2$ then $a^j_{i_1} \neq a^j_{i_2}$,
3. if $a^j_i = a^k_l$ then $n^j_i \equiv n^k_l$.

□

Next we show that any finite process is provably equal to a normal form.

Lemma 4.12 (Normalization) *For all $d \in FinProc$ there is a normal form $nf(d)$ such that $d =_E nf(d)$.*

Proof Let $dpth(d) = max\{|\sigma| \,|d \stackrel{\sigma}{\Longrightarrow}\}$. We prove the statement by induction on $dpth(d)$. The base case follows from an easy structural induction on d. To prove the inductive step we also proceed by structural induction on d.

$d \equiv \Theta, \Omega$: Obvious.

$d \equiv a.d_1$: By induction there is a normal form, $nf(d_1)$, such that $nf(d_1) =_E d_1$. Then $a.nf(d_1) =_E a.d_1$, where $a.nf(d_1)$ is a normal form.

$d \equiv d_1 \oplus d_2$: By structural induction $d_1 =_E n_1$ and $d_2 =_E n_2$ where n_1 and n_2 are normal forms. If either $n_1 \equiv \Omega$ or $n_2 \equiv \Omega$, then $d_1 \oplus d_2 =_E \Omega$ where Ω is a normal form. Also the case where $n_i \equiv \Theta$ for $i = 1$ or $i = 2$ is trivial. So assume

$$n_1 \equiv \sum_{j \leq N} \sum_{i \leq N_j} a_i^j.n_i^j \text{ and } n_2 \equiv \sum_{l \leq M} \sum_{k \leq M_l} b_k^l.m_k^l.$$

Then

$$\begin{array}{l} d_1 \oplus d_2 =_E n_1 \oplus n_2 =_E \\ \sum_{j \leq N} \sum_{i \leq N_j} a_i^j.n_i^j \oplus \sum_{l \leq M} \sum_{k \leq M_l} b_k^l.m_k^l =_E \\ \sum_{s \leq N+M} d_s, \end{array}$$

where $d_s \equiv \sum_{i \leq N_s} a_i^s.n_i^s$ for $s \leq N$ and $d_{N+s} \equiv \sum_{k \leq M_s} b_k^s.m_k^s$ for $s \leq M$. Obviously condition 1. and 2. for normal forms hold. To ensure 3. we apply the equation (Der) of Lemma 4.4 repeatedly. Now condition 2. and 3. hold but 1. may not hold any more. However now we may apply the outer induction hypothesis to rewrite the term to normal form.

$d \equiv d_1 + d_2$: By structural induction, for $i = 1, 2$, there is a normal form n_i such that $d_i =_E n_i$. This implies $d =_E n_1 + n_2$. To rewrite $n_1 + n_2$ into a normal form, using E, we proceed as follows: First we distribute $+$ over $\oplus$, i.e. apply $(+ \oplus 1)$, sufficiently often so we end up with a double sum of the form $\sum_s \sum_r \alpha_r^s.\eta_r^s$. Then we apply equation $(pre + \oplus 1)$ to obtain 2. Now we may proceed as in the previous case.

□

The next step of the completeness proof is to prove the completeness for normal forms.

Lemma 4.13 (Partial Completeness) *For all normal forms n, m*

$$n \preceq m \text{ iff } n \sqsubseteq_E m.$$

Proof The "if" part of the statement follows from the soundness of the proof system. Therefore we only have to prove the "only if" part. We proceed by structural induction on n. If $n \equiv \Omega$ we are done. Also the case $n \equiv \Theta$ follows easily. So assume $n \equiv \sum_{j\leq N} \sum_{i\leq N_j} a_i^j.n_i^j$. As $n \downarrow$ we have that $m \downarrow$ and has the same form, $m \equiv \sum_{l\leq M} \sum_{k\leq M_l} b_k^l.m_k^l$. Let $j \leq N$. As $n \preceq m$ there is an $l_j \leq M$ such that $\{a_1^j, \ldots, a_{N_j}^j\} = \{b_1^{l_j}, \ldots, b_{M_{l_j}}^{l_j}\}$. This means that we may assume that $Nj = M_{l_j}$ and $a_i^j = b_i^{l_j}$ for $j \leq N$. Because of the structure of the normal forms $Der(a_i^j, Stb(n)) = Stb_w(n_i^j)$ and $Der(b_i^{l_j}, Stb(m)) = Stb_w(m_i^{l_j}), i \leq N_j$. This implies

$$Stb_w(n_i^j) \preceq Stb_w(m_i^{l_j}),$$

or equivalently that $n_i^j \preceq m_i^{l_j}$. By induction $n_i^j \sqsubseteq_E m_i^{l_j}$. This implies

$$\sum_{i\leq N_j} a_i^j.n_i^j \sqsubseteq_E \sum_{k\leq M_{l_j}} b_k^{l_j}.m_k^{l_j}. \tag{3}$$

In a similar way we may show that for all $l \leq M$ there is a $j_l \leq N$ such that

$$\sum_{i\leq N_{j_l}} a_i^{j_l}.n_i^{j_l} \sqsubseteq_E \sum_{k\leq M_l} b_k^l.m_k^l. \tag{4}$$

From (3) and (4) we get

$$\begin{aligned} n \equiv\ & \textstyle\sum_{j\leq N} \sum_{i\leq N_j} a_i^j.n_i^j =_E \\ & \textstyle\sum_{j\leq N} \sum_{i\leq N_j} a_i^j.n_i^j \oplus \sum_{l\leq M} \sum_{i\leq N_{j_l}} a_i^{j_l}.n_i^{j_l} \sqsubseteq_E \\ & \textstyle\sum_{j\leq N} \sum_{k\leq M_l} b_k^{l_j}.m_k^{l_j} \oplus \sum_{l\leq M} \sum_{k\leq M_l} b_k^l.m_k^l =_E m. \end{aligned}$$

This completes the proof of the theorem. □

Finally we show the completeness for the full language $Proc$.

Theorem 4.14 (Completeness) *For all $p, q \in Proc$*

$$p \preceq q \text{ iff } p \sqsubseteq_{E_{rec}} q$$

Proof The "if" implication follows from the soundness of the proof system. For the "only if" part we proceed as follows:

$$\begin{aligned} p \preceq q \ &\Rightarrow \forall n.p^n \preceq q && \text{(by Lem. 4.3 and 4.2)} \\ &\Rightarrow \forall n \exists m.p^n \preceq q^m && \text{(by Cor. 4.9)} \\ &\Rightarrow \forall n \exists m.nf(p^n) \preceq nf(q^m) && \text{(by Lem. 4.12)} \\ &\Rightarrow \forall n \exists m.nf(p^n) \sqsubseteq_E nf(q^m) && \text{(by Lem. 4.13)} \\ &\Rightarrow \forall n \exists m.p^n \sqsubseteq_E q^m && \text{(by Lem. 4.12)} \\ &\Rightarrow \forall n.p^n \sqsubseteq_{E_{rec}} q && \text{(by Lem. 4.3)} \\ &\Rightarrow p \sqsubseteq_{E_{rec}} q && \text{(by the rule } (\omega)\text{).} \end{aligned}$$

Here $nf(p^n)$ has the same meaning as in the normalization Theorem 4.12, i. e. $nf(p^n)$ is a normal form where $nf(p^n) =_E p^n$. □

5 Conclusion

In this study we have given three characterizations of the readiness semantics, one by means of lighted button pressing experiments, a bisimulation like one and a modal characterization given some restrictions on the underlying LTS. The most important of these restrictions is the assumption that the communication capabilities are preserved by internal transitions which is not true for standard process description languages like CCS. We have applied the theory on a concrete language that has this property, the regular sublanguage of $CCS_{\oplus}$, CCS without τs. For this language we also characterized the readiness semantics by means of a sound and complete proof system. We used the modal characterization to prove the algebraicity of the behavioural preorder and thus reduced the completeness proof to a completeness proof over recursion free terms.

To complete this study we would like to be able to reason about the full language $CCS_{\oplus}$. As pointed out in the introduction, if we add the parallel operator with the standard operational semantics to the language, the internal actions do not preserve communication capabilities. A future task is therefore to investigate whether the parallel operator may be modelled suitably so it fits into this set–up. Another obvious extension of this work is to use similar methods to obtain the same kind of characterization of the failures semantics [OH86] and the must testing [DNH84]. Most of the work is already done by the author and will be recorded elsewhere.

References

[Abr91] S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92:161–218, 1991.

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

[BKO88] J.A. Bergstra, J.W. Klop, and E.-R. Olderog. Readies and failures in the algebra of communicating processes. *SIAM Journal on Computing*, 17(6):1134–1177, 1988.

[BM92] B. Bloom and A.R. Meyer. Experimenting with process equivalence. *Theoretical Computer Science*, 101(2):223-237, 1992.

[DNH84] DeNicola, R. and M. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83-133, 1984.

[Gla90] R.J. van Glabbeek. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90, Amsterdam*, volume 458 of *Lecture Notes in Computer Science*, pages 278–297. Springer-Verlag, 1990.

[Gla93] Robert Jan van Glabbeek. The linear time – branching time spectrum II: the semantics of sequential processes with silent moves. In E. Best, editor, *Proceedings CONCUR 93, Hildesheim, Germany*, volume 715 of *Lecture Notes in Computer Science*, pages 66–81. Springer-Verlag, 1993.

[Hen88] M. Hennessy. *Algebraic Theory of Processes*. MIT Press, Cambridge, Massachusetts, 1988.

[HI93] M. Hennessy and A. Ingólfsdóttir. A theory of communicating processes with value-passing. *Information and Computation*, 107(2):202–236, 1993.

[Ing95] A. Ingólfsdóttir. A semantic theory for value-passing processes late approach — Part II: A behavioural semantics and full abstractness. Report RS–95–22, BRICS (Basic Research in Computer Science, Centre of the Danish National Research Foundation), Institute for Electronic Systems, Department of Mathematics and Computer Science, Aalborg University Centre, April 1995.

[Knu73] D.E. Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, 1973. Second edition.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

[OH86] E.-R. Olderog and C.A.R. Hoare. Specification-oriented semantics for communicating processes. *Acta Informatica*, 23:9–66, 1986.

[SI94] Bernhard Steffen and Anna Ingólfsdóttir. Characteristic formulae for processes with divergence. *Information and Computation*, 110(1):149–163, April 1994.

[Tar55] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5, 1955.

A Proof of Theorem 2.6

First we state the following results.

Lemma A.1 *Let $s, t \in State, \sigma \in Lab^*$ and $l \in Lab$. If $s \downarrow l$ and $s \stackrel{l\sigma}{\Longrightarrow} t \not\longrightarrow$ then $s \stackrel{\varepsilon}{\Longrightarrow} s' \stackrel{l}{\Longrightarrow} t' \stackrel{\sigma}{\Longrightarrow} t$ for some $s', t' \not\longrightarrow$.*

Proof Follows easily from Assumption 2.2. □

In what follows we assume that the set Lab is infinite. To prove Theorem 2.7 first we review the definition of the readiness semantics as it is given in [OH86]. The set of observations Obs is given by

$$Obs = \{\sigma, \sigma A | \sigma \in Lab^*, A \subseteq_{fin} Lab\}.$$

Then the set of observations of a state s, $obs(s)$ is defined as the least set satisfying

1. $s \stackrel{\sigma}{\Longrightarrow}$ implies $\sigma \in obs(s)$,
2. $s \stackrel{\sigma}{\Longrightarrow} s' \not\longrightarrow$ and $I(s') = A$ implies $\sigma A \in obs(s)$,
3. $s \uparrow \sigma$ implies $\sigma, \sigma A \in obs(s)$ for all $A \subseteq_{fin} Lab$.

Then s and t are readiness equivalent iff $obs(s) = obs(t)$. Now let

$$Obs^* = \{\sigma A | s \in Lab^*, A \subseteq_{fin} Lab\}.$$

and $obs^*(s)$ be defined as the least set satisfying 2. and 3. in the definition of obs. Then we have

Lemma A.2 *For all $s, t \in S$ and $\sigma \in Lab^*$,*

1. *$obs^*(s) = obs^*(t)$ implies $(\forall \sigma. s \downarrow \sigma \;\Leftrightarrow\; t \downarrow \sigma)$,*
2. *$obs(s) = obs(t)$ iff $obs^*(s) = obs^*(t)$.*

Proof

1. This may be proved as follows:

   Assume $obs^*(s) = obs^*(t)$ and $s \downarrow \sigma$. We have the following cases:

   $s \stackrel{\sigma}{\not\Longrightarrow}$: Then $\sigma A \notin obs^*(s) = obs^*(t)$ for all A. This in turn implies $t \downarrow \sigma$.

$s \stackrel{\sigma}{\Longrightarrow}$: Let $\mathcal{A} = \{I(s') | s \stackrel{\sigma}{\Longrightarrow} s' \not\longrightarrow\}$. By Lemma 2.3, $\mathcal{A}$ is finite and as Lab is infinite we may choose a $A \subseteq_{fin} Lab$ such that $A \notin \mathcal{A}$. Now it is easy to see that $\sigma A \notin obs^*(s)$ and therefore, by our assumption, $\sigma A \notin obs^*(t)$. This in turn implies that $t \downarrow \sigma$.

The statement now follows by symmetry.

2. Now we proceed as follows:

"only if": Follows immediately.

"if": Assume $obs^*(s) = obs^*(t)$. By symmetry it is sufficient to prove that whenever $o \in obs(s)$ then $o \in obs(t)$. The only non-trivial case is when $o = \sigma$. So assume $\sigma \in obs(s)$. If $s \uparrow \sigma$ then, by part 1 of this lemma, $t \uparrow \sigma$ and therefore $\sigma \in obs(t)$. Next assume $s \downarrow \sigma$. Then $s \stackrel{\sigma}{\Longrightarrow} s'$ where $s' \not\longrightarrow$. This implies

$$\sigma I(s') \in obs^*(s) = obs^*(t) \subseteq obs(t),$$

which in turn implies $\sigma \in obs(t)$.

□

To prove Theorem 2.7 it is now sufficient to prove that

$$s =_{\mathcal{R}} t \text{ iff } obs^*(s) = obs^*(t).$$

To obtain this first we extend the transition function $Der : Lab \times \mathcal{S} \hookrightarrow \mathcal{S}$ in the standard way to a transition function of the functionality $Der : Lab^* \times \mathcal{S} \hookrightarrow \mathcal{S}$ by

- $Der(\varepsilon, S) = S$,
- $Der(l\sigma, S) = S'$ iff $Der(l, S) = S''$ for some S'' where $Der(\sigma, S'') = S'$.

Sometimes we write $S \stackrel{\sigma}{\Longrightarrow} S'$ if $Der(\sigma, S) = S'$ and $S \stackrel{\sigma}{\Longrightarrow}$ if $S \stackrel{\sigma}{\Longrightarrow} S'$ for some S'. Also the convergence predicate $\downarrow$ is extended by

- $S \downarrow \varepsilon$ iff $S \downarrow$,
- $S \downarrow l\sigma$ iff $S \downarrow$ and $(l \in I(S) \;\; \Rightarrow \;\; Der(l, S) \downarrow \sigma)$.

We need the following intermediate results.

Lemma A.3 *$\forall s \in State, \sigma \in Lab^*, A \subseteq_{fin} Lab, S \in \mathcal{S}$.*

1. *$s \downarrow \sigma$ iff $Stb_w(s) \downarrow \sigma$,*

2. $S \downarrow \sigma$ *implies*

   (a) $S \stackrel{\sigma}{\Longrightarrow}$ *implies* $Der(\sigma, S) = \{s' | \exists s \in S. s \stackrel{\sigma}{\Longrightarrow} s' \wedge s' \not\longrightarrow\}$

   (b) $\{s' | \exists s \in S. s \stackrel{\sigma}{\Longrightarrow} s' \wedge s' \not\longrightarrow\} \neq \emptyset$ *implies* $S \stackrel{\sigma}{\Longrightarrow}$.

3.

   (a) $MayPass(\sigma A, Stb(s))$ *iff* $\sigma A \in obs^*(s) \wedge s \downarrow \sigma$

   (b) $MustFail(\sigma A, Stb(s))$ *iff* $\sigma A \notin obs^*(s)$.

Proof

1. This statement follows from the following two statements:

$$s \downarrow \sigma \text{ iff } (\forall s' \in Stb_w(s).\, s' \downarrow \sigma) \tag{5}$$

and

$$S \downarrow \sigma \text{ iff } (\forall s \in S.\, s \downarrow \sigma). \tag{6}$$

(where we use the convention $\perp \uparrow$). We prove each of these statements separately.

Statement (5):

"only if":We proceed by structural induction on σ. The base case $\sigma = \varepsilon$ follows immediately. For the inductive step assume $s \downarrow l\rho$ and $s' \in Stb_w(s)$ and we will prove that $s' \downarrow l\rho$. Obviously $s' \downarrow$. Next assume $s' \stackrel{l}{\Longrightarrow} s''$. Then $s \stackrel{l}{\Longrightarrow} s''$ and by assumption $s'' \downarrow \rho$. This implies $s' \downarrow l\rho$.

"if": Again we proceed by induction on σ and again the base case follows immediately. For the inductive step assume that for all $s' \in Stb_w(s)$, $s' \downarrow l\rho$. In particular $s' \downarrow$ for all $s' \in Stb_w(s)$ and by the base case, $s \downarrow$. So assume $s \stackrel{l}{\Longrightarrow} s''$. Then, by Lemma A.1, there is an $s^* \in Stb_w(s)$ such that $s^* \stackrel{l}{\Longrightarrow} s''$. By assumption $s'' \downarrow \rho$ which implies that $s \downarrow l\rho$.

Statement (6):

"only if": We proceed by induction in σ where the base cases $\sigma = \varepsilon, l$ follow immediately. For the inductive step, assume $S \downarrow l\rho$. Then $S \downarrow l$ and therefore $s \downarrow l$ for all $s \in S$ by the base case. In particular $s \downarrow$ for all $s \in S$. Next assume $s \in S$ and $s \stackrel{l}{\Longrightarrow} r'$, we will prove that $r' \downarrow \rho$. First we note that $l \in I(S)$ and $Der(l, S) \downarrow$. Furthermore $r' \downarrow$ and $Stb_w(r') \subseteq Der(l, S)$. By definition of $\downarrow \rho$ on $\mathcal{S}$, $Der(l, S) \downarrow \rho$ and by induction for all $r'' \in Der(l, S)$, $r'' \downarrow \rho$. In particular this is true for all $r'' \in Stb_w(r')$. By the "if" part of statement (5), $r' \downarrow \rho$ as we wanted to prove.

"if": Again we proceed by induction on σ and the base case follows immediately. So assume $s \downarrow l\rho$ for all $s \in S$. In particular this means that $s \downarrow$ for all $s \in S$ and therefore, by the base case $S \downarrow$. Next assume $l \in I(S)$, we will prove that $Der(l, S) \downarrow \rho$. To prove this assume $r \in Der(l, S)$. Then $s \stackrel{l}{\Longrightarrow} r$ for some $s \in S$. By assumption $r \downarrow \rho$. As this is true for any $r \in Der(l, S)$, by induction $Der(l, S) \downarrow \rho$, as wanted.

2. (a) Assume $S \downarrow \sigma$ and $S \stackrel{\sigma}{\Longrightarrow}$, i.e. $Der(\sigma, S)$ exists and is different from $\{\perp\}$ and $\emptyset$. Let

$$D(\sigma, S) = \{s' | \exists s \in S. s \stackrel{\sigma}{\Longrightarrow} s' \wedge s' \not\longrightarrow\}.$$

We will prove that $D(\sigma, S) = Der(\sigma, S)$ by induction on σ. The base cases $\sigma = \varepsilon$ and $\sigma = l$ follow immediately. For the inductive step, by definition of Der, by the case $\sigma = l$ and by induction we get

$$Der(l\sigma', S) = Der(\sigma', Der(l, S)) = Der(\sigma', D(l, S)) = D(\sigma', D(l, S)).$$

Therefore it only remains to prove that

$$D(\sigma', D(l, S)) = D(l\sigma', S).$$

To prove this first assume $s' \in D(l\sigma', S)$. This implies $s \stackrel{l\sigma'}{\Longrightarrow} s' \not\longrightarrow$ for some $s \in S$. By Lemma A.1, $s \stackrel{l}{\Longrightarrow} s''' \stackrel{\sigma'}{\Longrightarrow} s'$ for some $s''' \not\longrightarrow$, i.e. for some $s''' \in D(l, S)$. This implies that $s' \in D(\sigma', D(l, S))$.
Next assume $s' \in D(\sigma', D(l, S))$. This means that there is an $s'' \in D(l, S)$ such that $s'' \stackrel{\sigma'}{\Longrightarrow} s'$. As $s'' \in D(l, S)$, $s \stackrel{l}{\Longrightarrow} s''$ for some $s \in S$, i.e. $s \stackrel{l\sigma'}{\Longrightarrow} s'$ and therefore $s' \in D(l\sigma', S)$.

(b) We prove the statement by induction on σ. The base cases $\sigma = \varepsilon$ and $\sigma = l$ are obvious. For the inductive step assume that there are $s \in S$ and $s' \not\longrightarrow$ such that $s \stackrel{l\sigma}{\Longrightarrow} s'$. Then $s \stackrel{l}{\Longrightarrow} s'' \stackrel{\sigma}{\Longrightarrow} s'$ for some $s'' \not\longrightarrow$, i.e. for some $s'' \in Der(l, S)$. Furthermore $Der(l, S) \downarrow \sigma$. By induction $Der(l, S) \stackrel{\sigma}{\Longrightarrow}$. As $S \stackrel{l}{\Longrightarrow} Der(l, S)$ this implies $S \stackrel{l\sigma}{\Longrightarrow}$.

3. (a) The result follows from the following two statements:

$$MayPass(\sigma A, S) \text{ iff } S \downarrow \sigma, S \stackrel{\sigma}{\Longrightarrow} \text{ and } A \in \mathcal{I}(Der(\sigma, S)) \tag{7}$$

$$\begin{array}{l} Stb_w(s) \downarrow \sigma, Stb_w(s) \stackrel{\sigma}{\Longrightarrow} \text{ and } A \in \mathcal{I}(Der(\sigma, Stb_w(s))) \\ \text{iff } \ \sigma A \in obs^*(s) \wedge s \downarrow \sigma \end{array} \tag{8}$$

Statement (7):

"only if": We proceed by induction on σ where the base case is obvious. For the inductive step we proceed as follows. Assume $MayPass(l\sigma A, S)$, i.e. $S\downarrow, l \in I(S)$ and $MayPass(\sigma, Der(l,S))$. By induction

$$Der(l,S)\downarrow \sigma, Der(l,S) \stackrel{\sigma}{\Longrightarrow}$$

and

$$A \in \mathcal{I}(Der(\sigma, Der(l,S))) = \mathcal{I}(Der(l\sigma, S)).$$

This implies that $S\downarrow l\sigma$, $S \stackrel{l\sigma}{\Longrightarrow}$ and $A \in \mathcal{I}(Der(l\sigma,S))$.

"if": Again we proceed by induction on σ where the base case is obvious. For the inductive step we proceed as follows: Assume $S\downarrow l\sigma$, $S \stackrel{l\sigma}{\Longrightarrow}$ and $A \in \mathcal{I}(Der(l\sigma,S))$. Then

$$S \stackrel{l}{\Longrightarrow} Der(l,S) \stackrel{\sigma}{\Longrightarrow} Der(\sigma, Der(l,S)) = Der(l\sigma, S)$$

and $Der(l,S)\downarrow \sigma$. By induction $MayPass(\sigma A, Der(l,S))$. As $l \in I(S)$ and $S\downarrow$ this implies $MayPass(l\sigma A, S)$.

Statement (8):

"only if": Assume that

$$Stb(s)\downarrow \sigma,\ Stb(s) \stackrel{\sigma}{\Longrightarrow} \text{ and } A \in \mathcal{I}(Der(\sigma, Stb(s)).$$

By 1. of this lemma, $s\downarrow\sigma$. Furthermore $A \in \mathcal{I}(Der(\sigma, Stb(s))$ implies that $I(s') = A$ for some $s' \in Der(\sigma, Stb(s))$. By 2.$(a)$ of this lemma there is an $s'' \in Stb(s)$ such that $s'' \stackrel{\sigma}{\Longrightarrow} s'$, which implies $s \stackrel{\sigma}{\Longrightarrow} s'$. This shows that $\sigma A \in obs^*(s)$.

"if": Assume

$$\sigma A \in obs^*(s) \text{ and } s\downarrow\sigma.$$

Again by 1. of this lemma, $s\downarrow\sigma$ implies $Stb(s)\downarrow\sigma$. As $\sigma A \in obs^*(s)$ and $s\downarrow\sigma$ then $s \stackrel{\sigma}{\Longrightarrow} s' \not\longrightarrow$ for some s' where $I(s') = A$. By Lemma A.1, there is an $s'' \in Stb(s)$ such that $s'' \stackrel{\sigma}{\Longrightarrow} s'$. By 2.$(b)$ and (a) of this lemma, $Stb(s) \stackrel{\sigma}{\Longrightarrow}$ and $s' \in Der(\sigma, S)$. This implies that $A \in \mathcal{I}(Der(\sigma,S))$.

(b) May be proved in a similar way by first proving the following two statements

$$MustFail(\sigma A, S) \text{ iff } S\downarrow\sigma, S \stackrel{\sigma}{\Longrightarrow} \text{ and } A \notin \mathcal{I}(Der(\sigma,S)) \qquad (9)$$

$$S\downarrow\sigma, S \stackrel{\sigma}{\Longrightarrow} \text{ and } A \notin \mathcal{I}(Der(\sigma,S)) \text{ iff } \sigma A \notin obs^*(s). \qquad (10)$$

Here we note that $\sigma A \notin obs^*(s)$ implies $s\downarrow\sigma$.

□

Proof of Theorem 2.7

As pointed out before it is sufficient to prove that

$$obs^*(s) = obs^*(t) \text{ iff } s =_{\mathcal{R}} t.$$

We prove each implication separately.

"only if": Let $obs^*(s) = obs^*(t)$. First assume $MayPass(\sigma A, Stb_w(s))$, we will prove that $MayPass(\sigma A, Stb_w(t))$. By Lemma A.3.3 we have that $\sigma A \in obs^*(s) = obs^*(t)$ and $s \downarrow \sigma$. Lemma A.2.1 implies $t \downarrow \sigma$. Again by Lemma A.3.3. we may conclude that $MayPass(\sigma A, Stb(t))$. Similarly we prove that $MustFail(\sigma, Stb_w(s))$ implies $MustFail(\sigma, Stb_w(s))$ and the result follows by symmetry.

"if": Assume that $obs^*(s) \neq obs^*(t)$ witnessed by $\sigma A \in obs^*(s)$ but $\sigma A \notin obs^*(t)$. By Lemma A.3.3, $MustFail(\sigma A, Stb(t))$ and $\neg MustFail(\sigma A, Stb(s))$.

Recent Publications in the BRICS Report Series

RS-96-43 **Anna Ingólfsdóttir.** ***Weak Semantics Based on Lighted Button Pressing Experiments: An Alternative Characterization of the Readiness Semantics*****. November 1996. 36 pp. An extended abstract to appear in the proceedings of the** ***10th Annual International Conference of the European Association for Computer Science Logic*****, CSL '96.**

RS-96-42 **Gerth Stølting Brodal and Sven Skyum.** ***The Complexity of Computing the k-ary Composition of a Binary Associative Operator*****. November 1996. 15 pp.**

RS-96-41 **Stefan Dziembowski.** ***The Fixpoint Bounded-Variable Queries are PSPACE-Complete*****. November 1996. 16 pp. Presented at the** ***10th Annual International Conference of the European Association for Computer Science Logic*****, CSL '96.**

RS-96-40 **Gerth Stølting Brodal, Shiva Chaudhuri, and Jaikumar Radhakrishnan.** ***The Randomized Complexity of Maintaining the Minimum*****. November 1996. 20 pp. To appear in a special issue of** ***Nordic Journal of Computing*** **devoted to the proceedings of SWAT '96. Appears in Karlson and Lingas, editors,** ***Algorithm Theory: 5th Scandinavian Workshop*****, SWAT '96 Proceedings, LNCS 1097, 1996, pages 4–15.**

RS-96-39 **Hans Hüttel and Sandeep Shukla.** ***On the Complexity of Deciding Behavioural Equivalences and Preorders – A Survey*****. October 1996. 36 pp.**

RS-96-38 **Hans Hüttel and Josva Kleist.** ***Objects as Mobile Processes*****. October 1996. 23 pp.**

RS-96-37 **Gerth Stølting Brodal and Chris Okasaki.** ***Optimal Purely Functional Priority Queues*****. October 1996. 27 pp. To appear in** ***Journal of Functional Programming*****, 6(6), December 1996.**

RS-96-36 **Luca Aceto, Willem Jan Fokkink, and Anna Ingólfsdóttir.** ***On a Question of A. Salomaa: The Equational Theory of Regular Expressions over a Singleton Alphabet is not Finitely Based*****. October 1996. 16 pp.**