Axioms for Concurrency

Faron Moller

Doctor of Philosophy

University of Edinburgh

1989

Abstract

We study properties of equational characterisations of congruences defined over process algebras. The languages on which we concentrate are based on **CCS**, and our equivalences are generally restricted to observational congruences. We start by defining and investigating the notion of *extensionality* or *ω-completeness* of an axiomatisation with respect to some semantic equivalence, as an extension of simple completeness, and show that, whereas in some cases the ability to ω-completely axiomatise a system is straightforward, there are sometimes difficulties in doing this when our algebra contains the symmetric full merge operator.

We then consider the problem of decomposing a process into the parallel composition of simpler processes. Here we present several example systems where we can prove that any process term can be expressed uniquely as the parallel composition of *prime* process terms, those processes which cannot themselves be expressed as a parallel composition of at least two nontrivial processes.

We next consider the possibility of the nonexistence of finite axiomatisations for certain systems. In particular, we show that strong observational congruence over a subset of the usual **CCS** algebra with the full merge operator cannot be completely characterised by any finite set of equational axioms, thus requiring the power of the Expansion Theorem to present an infinite set of axioms within a single axiom schema. We then go on to prove that no reasonable stronger notion of congruence can be finitely axiomatised, thus explaining the difficulty presented in searching for complete laws for noninterleaving semantic congruences where the Expansion Theorem fails.

Finally we consider the same problems in an algebra containing a sequencing operator rather than the usual **CCS** action prefixing operator.

Acknowledgements

I would first like to record my gratitude to Robin Milner who demonstrated to me the recipe for a good supervisor: an abundant supply of inspiration, endless patience and encouragement, and a touch of gentle prodding when the results were slow coming.

Further inspiration, scholarly debate, and fun were always on offer from Mads Dam and Andreas Knobel. They were both willing sounding boards to my ideas, offering illuminating comments which almost always shaped my viewpoint and outlook on the problems addressed by this thesis.

After Robin and Mads and Andreas, my problems and ideas were then presented to the Wednesday afternoon Concurrency Club, where I was always guaranteed substantial and eager comments on the ideas, bringing them from half- to three-quarter-baked status. I thank all members of the Club who ever made a comment during one of my talks, as I am sure every comment made had its effect on the final product. I especially thank David Walker and Colin Stirling for extra discussions and interest taken in my ideas outside of the Club.

In the end, it was Chris Tofts who influenced the later stages of the work and took an early draft of my thesis on a skiing holiday.

Life was made enjoyable by numerous new friends I have made in Scotland, who made the experience of doing a thesis bearable. My thanks go out especially to Martin, Laurent, Lise, Murray and Ally. I am totally and forever thankful to Alice Dumas, my life-time friend, for her undying support.

This work was financially supported by a Scholarship from the Association of Commonwealth Universities.

Table of Contents

Chapter 1

Introduction

The need to be able to reason about computer programs in a rigorous formal way is self evident. This applies even more so to programs involving parallel constructs than to those written in a purely sequential programming language. This is due to the inherent intuitive difficulties discovered by experience posed by the nondeterministic behaviour of such programs. These methods for reasoning must furthermore be formal to allow verification proofs to be at least partly mechanised, as systems for which we desire to prove properties grow in size. Indeed there now exist several automated systems for reasoning about concurrent processes, just one example being the *Concurrency Workbench*[1], a prototype automated tool for reasoning about **CCS** agents, processes defined using Milner's *Calculus of Communicating Systems* [MIL80].

The motivation for this thesis is to add to the understanding of reasoning about concurrent processes using strictly equational logic. Rewrite rule based systems founded on equational theories are ideal for implementing in an automated tool, so a clear understanding of the equational theory of concurrent processes forms the basis of a rigorous proof technique amenable to automation. In this thesis we consider three basic problems of equational reasoning within process algebras.

---

[1]The *Concurrency Workbench* is currently under development in a joint SERC-funded project between the Laboratory for the Foundations of Computer Science at the University of Edinburgh and the University of Sussex.

Firstly, we analyse the possibility of reasoning about open terms (terms with free variables) in our process algebras. Such terms represent underspecified processes in which the indeterminates are placemarkers for unknown subprocesses which may for example be implemented by someone else, or may be irrelevant with respect to the property of processes which we are currently considering. Secondly, we consider the problem of decomposing processes into parallel components, and analyse when this can be done completely (that is, decomposed into nondecomposable components) in exactly one way. This would allow us to reason about the components of a system, and also allow us to find possibly the most optimal implementation of a specification with respect to the amount of parallelism which can be exploited in the implementation. Finally, we consider problems regarding the finite axiomatisability of certain systems. This study would lead us to deduce some finite presentation of a complete equational theory of a system, or more interestingly, to deduce when such a finite system cannot exist.

1.1 Extensional Completeness

A complete axiomatisation for reasoning about open terms of some process algebra with respect to some semantic congruence would amount to what is referred to for example in the lambda calculus as a complete *extensional* theory, or an *ω-complete* axiomatisation. Such an extensional theory for CCS would allow us to reason about underspecified concurrent systems without having to resort to tactics other than pure equational reasoning, such as some form of induction. This would almost certainly give much simpler proofs to many valid open statements in CCS than can be derived within a simply complete axiomatisation. For example, using the usual complete axiomatisation for finite CCS terms, the proofs of the associativity and commutativity of the parallel operator are less than straightforward to prove, involving lengthy proofs by induction (see, *e.g.,* [MIL80], **Theorem 5.5**). An automated system for reasoning about CCS processes based on an extensional axiomatisation would neither need to restrict itself to *agents* (closed term processes), nor need to resort to any tactic other than pure equational reasoning.

1.2 Unique Decompositions

If it were the case that in some system for reasoning about concurrent processes we could prove the validity of a unique decomposition theorem, then this would provide two points of interest. Firstly, it would gives us a start towards a normal form for processes, leading towards the ability to prove certain completeness results. More importantly, it would allow us to reason about the maximal degree of parallelism which exists within a system.

For example, consider a language over the signature $\Sigma = \{\mathbf{0}, ., +, \| \}$, consisting of the usual **CCS** operators representing the null process, action prefix, nondeterministic choice and full merge. A normal form for process terms in this language could be defined to be

$$\prod_{1 \leq i \leq n} P_i \qquad (n \geq 0),$$

where each P_i represented a prime process and as such is in some form of *prime normal form.* A prime process over this language would be one which could not be expressed as the parallel composition of two or more nontrivial processes. Upon performing some transition, the term may evolve into a nonprime process, but it must move from its state in order to express any global concurrency. As such, prime normal forms could be defined to be

$$\sum_{1 \leq i \leq n} a_i.p_i \qquad (n > 0),$$

where each p_i was in normal form. Such prime normal forms would of course be restricted to prime processes; for instance, $a.b.\mathbf{0} + b.a.\mathbf{0}$ would not be a prime normal form term as it is not prime, being equal to the composite term $a.\mathbf{0} \parallel b.\mathbf{0}$.

Given that our congruence over this signature allows for a unique decomposition theorem to hold, any term could be expressed uniquely as a normal form as defined above. Moreover, we could define the maximum amount of parallelism of an agent as follows: Given a process P expressed in normal form by

$$P = \prod_{1\leq i\leq m}\Big(\sum_{1\leq j\leq n_i} a_{ij}.p_{ij}\Big),$$

the maximum amount of parallelism which inherently exists within P would be given by

$$\mathrm{maxpar}(P) = \sum_{1\leq i\leq m}\Big(\max_{1\leq j\leq n_i}\mathrm{maxpar}(p_{ij})\Big),$$

where here we let $\sum_{i\in\emptyset} i = 1$. Such a definition could have implications on the implementation of a process specified by a **CCS** agent; it would reveal to the implementor how much parallelism he could attain in implementing the abstract process, telling him how many actual control processes he could use to maximise efficiency.

It is not in general the case that such a decomposition result is possible. For instance, if the operational semantics of the full merge operator obeys the following **CSP** rule (*cf.* [BRO84], [HOA85])

$$\frac{P \xrightarrow{a} P', \quad Q \xrightarrow{a} Q'}{P \parallel Q \xrightarrow{a} P' \parallel Q'},$$

then we can see that the possibility of a unique decomposition fails immediately, as

$$\begin{aligned} a.\mathbf{0} &= a.\mathbf{0} \parallel a.\mathbf{0} \\ &= a.\mathbf{0} \parallel a.\mathbf{0} \parallel \cdots \parallel a.\mathbf{0}. \end{aligned}$$

Also, as pointed out by R. van Glabbeek ([KLO87]), a unique factorisation result could not hold for certain *failure* or *readiness* semantics, where we have the law

$$a.(b.x + b.y) \;=\; a.b.x \;+\; a.b.y,$$

for in this case the process term $a.a.\mathbf{0} + a.a.a.\mathbf{0} + a.a.a.a.\mathbf{0}$ would have two distinct decompositions, namely

$$\begin{aligned} & a.a.\mathbf{0} + a.a.a.\mathbf{0} + a.a.a.a.\mathbf{0} \\ &\quad = (a.\mathbf{0} + a.a.\mathbf{0}) \parallel (a.\mathbf{0} + a.a.\mathbf{0}) \\ &\quad = a.\mathbf{0} \parallel (a.\mathbf{0} + a.a.\mathbf{0} + a.a.a.\mathbf{0}). \end{aligned}$$

However, in this thesis we show that the result does hold for the above subset of **CCS** under the usual operational laws for the operators (with or without communication via synchronisation of complementary actions allowed with the parallel operator), with respect to strong and weak observational congruence.

1.3 Finite Axiomatisability

One of the major problems which this thesis addresses is that of the nonexistence of finite equational axiomatisations for concurrent systems. In particular, we show that the above subset of **CCS** is not finitely axiomatisable under the semantic equivalence of strong observational congruence. The implications of such a result are straightforward: in order to completely axiomatise the congruence within the system, we need to find some (hopefully elegant) valid axiom schemata in order to represent an infinite set of axioms. In the case of strong congruence, the Expansion Theorem of **CCS** provides just what is needed.

We however manage to extend our result to apply over the same language under any *reasonable* congruence stronger than strong observational congruence. Thus we show that if we had some reasonable (defined in a strict, formal sense) noninterleaving semantic congruence, we would need to find some alternative axiom schemata to replace the Expansion Theorem, which would no longer hold in our stronger equivalence.

An interesting point about our nonexistence proofs is that they necessarily employ a method unlike that usually used for such results. The typical style of proof for this type of result is of a model-theoretic nature, using the following technique: suppose we have some infinite set $\mathcal{T}$ of axioms which completely characterises our notion of congruence, and that this theory can be expressed as $\mathcal{T} = \bigcup_{i \geq 0} \mathcal{T}_i$, where $\mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \cdots$ is an increasing chain of theories approximating $\mathcal{T}$. Suppose further that for each $k \geq 0$ there is a statement S_k such that $\mathcal{T} \vdash S_k$, but $\mathcal{T}_k \nvdash S_k$. Then there cannot exist a finite axiomatisation for our congruence. For suppose that $\mathcal{F}$ is a finite set of valid axioms which also characterises our congruence.

Then we would have that $\mathcal{F} \vdash S_k$ for each $k \geq 0$. From the validity of the axioms in $\mathcal{F}$, we could deduce that $\mathcal{T} \vdash \bigwedge \mathcal{F}$, and so (from the compactness theorem) that $\mathcal{T}' \vdash \bigwedge \mathcal{F}$ for some finite $\mathcal{T}' \subseteq \mathcal{T}$. But then there would be a k such that $\mathcal{T}' \subseteq \mathcal{T}_k$, and so we would have that $\mathcal{T}_k \vdash \mathcal{T}' \vdash \bigwedge \mathcal{F} \vdash S_k$, so by monotonicity we would have that $\mathcal{T}_k \vdash S_k$, contradicting $\mathcal{T}_k \nvdash S_k$.

Using the above technique, we must be careful about one point in particular. In order for our complete theory $\mathcal{T}$ to imply each of the axioms in $\mathcal{F}$, we must be sure that our theory $\mathcal{T}$ is not simply complete with respect to our congruence, but in fact complete for the extensional theory. In view of the fact that such a complete set of laws is not discovered in this thesis for the language in which we are interested, we could not apply the above model-theoretic argument. Hence all of our nonexistence of finite axiomatisation results are proven using a proof-theoretic strategy, rather than the usual model-theoretic method outlined above.

1.4 Layout of the Thesis

In the remainder of this introduction, we summarise the work presented in the thesis.

In **Chapter 2**, we present the necessary background material to our study. In particular, we describe the framework for the process algebras in which we shall be interested, as well as the equivalences between processes with which we shall work. Finally we fix some fundamental notions about equational proof systems in order to prove results about equational provability.

In **Chapter 3**, we investigate the extensional theories for strong observational congruence for different subsets of **CCS**. In particular, we derive a complete set of equations for extensional reasoning about a simple language of nondeterministic agents, followed by a complete extensional axiomatisation for a language containing the full and left merge operators, thus allowing concurrent computations. Finally, we present the difficulties involved in trying to do the same for the subset

of CCS containing the full merge operator in the absence of the simplifying left merge operator, and leave the problem of its extensional axiomatisation unsolved.

In **Chapter 4**, we investigate the unique decomposability of CCS agents. We first prove that the unique decomposition theorem is valid for a small subset of CCS with respect to strong observational congruence when communication is prohibited. We actually present two proofs of this result, the first originally presented by Milner and used as a model for the more difficult cases to follow, and the second a much simpler proof which works only in this basic framework. We then proceed on to show that the factorisation result remains valid when we allow communication, and also when we abstract away from internal communications by considering weak observational congruence.

In **Chapter 5**, we present two major results on the nonexistence of finite equational axiomatisations for process algebras. The first result shows that strong observational congruence over the above subset of CCS cannot be finitely axiomatised, thus showing the necessity of some axiom schemata such as that presented by the Expansion Theorem. Before going on to the second major result, we comment on the applicability of the proof to the same language when communication is allowed by the parallel operator. We then extend the result to show that no reasonable congruence which is stronger than strong observational congruence can be finitely axiomatised, thus posing the problem of finding applicable axiom schemata when trying to characterise a noninterleaving semantic congruence where the Expansion Theorem is no longer valid.

In **Chapter 6**, we consider the same problems for a different process algebra, a subset of ACP, which utilises a sequential composition operator in place of the CCS action prefix operator. We first introduce the **0** process into this framework in a manner different from, and seemingly more natural with respect to observational behavioural semantics than that of the researchers in the Dutch school who are the innovators and main proponents of this type of process algebra. We do however define their congruence over this language with the **0** process and characterise it as an observational congruence. We then proceed to examine our semantic congruence, presenting a finite equational axiomatisation for closed

term reasoning for the nondeterministic language, and present problems with its extensional axiomatisability. Finally, we extend the proof of the previous chapter to show that this language with the full merge operator added cannot be finitely axiomatised.

In **Chapter 7**, we present a short summary of our results, and outline some of the problems which we could not solve but whose solutions would find their rightful place in the main body of this thesis.

Chapter 2

Background

In this chapter, we lay the groundwork for our study. We present the languages in which we shall be interested, as well as the behavioural equivalences which we shall be studying. We also present relevant definitions and known results concerning our languages and equivalences. Finally, we present properties of equational characterisations which we shall be studying.

2.1 Process Algebras

Almost all of the languages which we shall consider will be sublanguages of that defined by the signature given in **Figure 2–1**, presented as a two-sorted algebra in the style of [EHR85]. We first presuppose a *nonempty* set of atomic action symbols **Act**, as well as a set of process variables **Var**. Then the languages which we shall consider will be defined to be the least sets of terms containing the variables **Var**, and closed under different subsets of the term-building operators given in the signature of **Figure 2–1**.

This set of operators is derived from a subset of those in the pure **CCS**. However, the usual parallel operator has been split into two separate operators, representing whether or not communication can occur between concurrent processes. As a way of introduction to the operators in the full signature, we give here some remarks on each of them.

Sorts	$\mathcal{P}$			(processes)
	Act			(atomic actions)
Operators	**0**	:	$\mathcal{P}$	(null program)
	.	:	$\mathbf{Act} \times \mathcal{P} \to \mathcal{P}$	(action prefix)
	$+$	:	$\mathcal{P} \times \mathcal{P} \to \mathcal{P}$	(nondeterministic choice)
	$\parallel$	:	$\mathcal{P} \times \mathcal{P} \to \mathcal{P}$	(full merge)
	$\lfloor\!\lfloor$	:	$\mathcal{P} \times \mathcal{P} \to \mathcal{P}$	(left merge)
	$\mid$	:	$\mathcal{P} \times \mathcal{P} \to \mathcal{P}$	(parallel)
	μ	:	$(\mathcal{P} \to \mathcal{P}) \to \mathcal{P}$	(recursive definition)

Figure 2–1: CCS-like operators

1. **0** is a nullary operator representing the null process, one which can perform no observable action, but simply terminates.

2. . is a binary operator which given an action symbol $a \in \mathbf{Act}$ and a process p, returns the process which can perform the action a and evolve into the process p upon so doing.

3. $+$ is a binary operator which given two processes, returns the process which is capable of choosing between the two and observably behaving exactly as the chosen process.

4. $\parallel$ is a binary operator which given two processes, returns the process which is capable of performing the actions of each of the processes, in the order which the two processes would have performed them, in an arbitrary interleaved fashion.

5. $\lfloor\!\lfloor$ is a binary operator which given two processes, returns the process which is capable of performing the actions of the two processes concurrently as with the interleaving operator $\parallel$ above, but with the stipulation that the first process (the left operand) must contribute the first action.

6. $|$ is the usual CCS binary parallel operator which given two processes, returns the process which is capable of performing the actions of the interleaved processes as defined for the full merge operator $\|$ above, but also at any time performing an action τ representing the synchronisation of complementary actions, each being offered at once by the two processes involved, and then evolving into the same concurrent composition of the resulting processes.

7. μ is a unary operator which given a function from processes to processes represented by a pair $(x.t(x))$, consisting of a variable x and a process term $t(x)$ with x (possibly) appearing *free* (*i.e.,* not within the scope of another μ operator involving the same variable x), returns the process P which represents a particular solution to the equation $P = t(P)$.

When writing out terms from some subset of this signature, we shall use parentheses to allow for unambiguous parsing. However, we shall minimise their use by allowing the action prefix operator to take precedence over all of the concurrency operators, which in turn will take precedence over the recursion operator, which finally in turn will take precedence over the summation operator. Furthermore, for the sake of economy in writing, we shall usually omit occurrences of the action prefix operator, and occurrences of $\mathbf{0}$ at the ends of action-prefixed subterms, thus for example rendering $a.b.\mathbf{0} \mid c.\mathbf{0}$ as $ab \mid c$.

Notice that we have not included either of *renaming* or *restriction* from pure CCS in our languages. This was simply because the problems of axiomatisations which we wish to address in this thesis arise without these operators, and adding them is usually not a problem with respect to axiomatisability; these operators are generally easily dealt with via their distributivity properties. Furthermore, we shall make little use of the recursion operator μ. The sole purpose of including it in the above signature is so that we can relate a particular result of Milner's on regular (finite-state) behaviours to our work. For the rest of the thesis, we shall work solely with finite CCS terms. This is as we are interested firstly in complete axiomatisations for our languages, which is not possible in general in the case of recursive terms using the parallel construct, and secondly in finite axiomatisations,

which seems likely not to be the case with regular behaviours (due to the results of *e.g.*, [CON71]).

More importantly, we allow the set **Act** to be a parameter in the definition of our languages, and consider our systems to be defined as two-sorted signatures. This is in contrast to the usual approach in **CCS**, where we only deal with a single sort $\mathcal{P}$, that consisting of our process terms. In that case, action prefix is considered as defining a set of unary operators $\{a. \mid a \in \mathbf{Act}\}$, one for each available action symbol. This adds complications to axiomatisations which we want to avoid. For instance, in what follows we shall wish to prove some results regarding the nonexistence of finite equational axiomatisations of certain congruences over our languages. In particular, we shall want to show that over a simple subset of finite **CCS** terms, the Expansion Theorem is not replaceable by any finite set of equational axioms. If we took the usual view of **CCS**, with an infinite action set **Act** this result would be almost immediate, as in order to express all that is expressed by the Expansion Theorem, each of the infinite number of unary action prefix operators would have to be explicitly mentioned. For instance, consider the following instance of the Expansion Theorem:

$$\alpha \parallel \beta = \alpha\beta + \beta\alpha.$$

This is actually an axiom schema representing an infinite number of axioms (when **Act** is infinite), where α and β are metavariables ranging over the set **Act**. In our formulation, α and β above are simply variables of sort **Act**, and the above represents a single axiom containing α and β as variables. More strongly than this though, our result will say that the Expansion Theorem cannot be replaced by a finite set of axioms even if the action set **Act** is finite, indeed even if it is a singleton set. Thus our proof will say that the Expansion Theorem is somehow inherently not finitely axiomatisable.

For a given process language $\mathcal{P}$, we shall distinguish the sublanguage $\mathcal{P}^0 \subseteq \mathcal{P}$ consisting of the *closed* terms in $\mathcal{P}$, or what Milner calls *agents*. These are the terms which have no free process variables, where the set of free process variables of a term is specified by the following definition.

Definition 2.1.1 *The set of* free process variables *of a term t, denoted by $fv(t)$, is defined structurally as follows:*

$$\begin{aligned} fv(x) &= \{x\}, \\ fv(0) &= \emptyset, & fv(t \parallel u) &= fv(t) \cup fv(u), \\ fv(at) &= fv(t), & fv(t \lfloor\!\lfloor u) &= fv(t) \cup fv(u), \\ fv(t+u) &= fv(t) \cup fv(u), & fv(t \mid u) &= fv(t) \cup fv(u), \\ & fv(\mu x.t) = fv(t) \setminus \{x\}. \end{aligned}$$

Thus we have the set of agents of $\mathcal{P}$ defined by:

$$\mathcal{P}^0 = \{t \in \mathcal{P} \mid fv(t) = \emptyset\}.$$

Any process variable which occurs free in a term $t \in \mathcal{P}$ (*i.e.*, not bound by the μ operator) represents some unspecified process which can be replaced by some agent in the language at any time. We shall thus define the *substitution* of terms for variables accordingly by the following definition.

Definition 2.1.2 (Substitution) *Given a term t and a substitution σ which is simply a mapping from some finite subset $\mathbf{V}$ of $\mathbf{Var}$ to $\mathcal{P}$, specified as a set $\{p_x/_x \mid x \in \mathbf{V}\}$, we define $t\sigma$ to be the term t with the occurrences of the free variables appearing in $\mathbf{V}$ replaced by the terms to which they are mapped by σ.*

All of the equivalences $\sim$ which we define shall be defined initially for closed terms only. However, we shall extend these equivalences to apply to open terms as well by defining $t \sim u$ *iff* for all closed substitutions σ defined over the set $fv(t) \cup fv(u)$, we have that $t\sigma \sim u\sigma$. It is clear that the resulting relation is conservative (the new relation restricted to $\mathcal{P}^0$ is precisely the equivalence with which we started), that it is an equivalence relation on all of $\mathcal{P}$, and that if the original equivalence was in fact a *congruence* on $\mathcal{P}^0$ (*i.e.*, was a substitutive equivalence relation), then the new relation itself will be a congruence on the whole of $\mathcal{P}$.

2.2 Equivalences Between Processes

2.2.1 Transition Systems

The equivalences in which we shall be interested for our languages will be *observational equivalences* as defined by Milner and Park (see *e.g.,* [MIL80], [PAR81]) via the notion of *bisimulations.* These equivalences are defined with respect to a *labelled transition system,* a general model of computation described in [KEL76] and used extensively in the study of **CCS**-like languages for defining the operational behaviour of processes. For a given process algebra $\mathcal{P}$ parameterised by an action set **Act**, a labelled transition system is a relation $\longrightarrow \subseteq \mathcal{P} \times \mathbf{Act} \times \mathcal{P}$ which formally defines what atomic actions a term in our language $\mathcal{P}$ is capable of performing, and what new processes will evolve upon performing particular actions: $P \xrightarrow{a} P'$ means that the process P may perform the action a and evolve into the process P' upon so doing.

For a particular language $\mathcal{P}$, the transition system which we define will satisfy the subset of rules laid out in **Figure 2–2** which pertain to the operators used in the algebra of $\mathcal{P}$. These rules give a *structural operational semantics* to our process languages, a natural method of presenting such semantics first proposed in [PLO81]. The rules are specified in an inferential style; if the sentence(s) mentioned above a line hold(s), then the sentence below the line must also hold. For instance, if the signature of our language includes the operators ., + and $\|$, then from rule (1) we would have that $b \xrightarrow{b} \mathbf{0}$, so by rule (3$b$) we would have that $a \,\|\, b \xrightarrow{b} a \,\|\, \mathbf{0}$.

We in fact want to allow a transition to be valid when and only when it can actually be derived from the rules laid out in **Figure 2–2**. Hence for a particular language $\mathcal{P}$ defined by a subset of the complete signature of **Figure 2–1**, we define our transition system to be the *least* relation $\longrightarrow \subseteq \mathcal{P} \times \mathbf{Act} \times \mathcal{P}$ satisfying the subset of rules which are relevant (that is, those rules involving the operators in the signature of $\mathcal{P}$). This defines the operational behaviour of our processes.

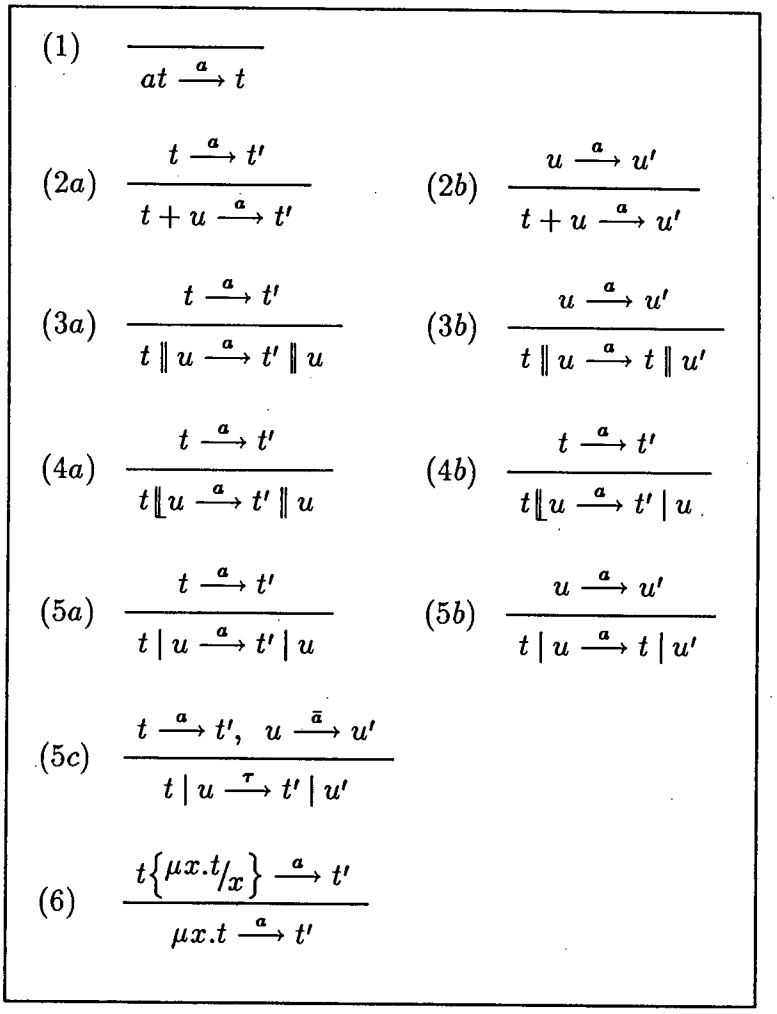

Figure 2–2: Operational rules for **CCS** operators.

We can extend this definition to apply to *sequences* of atomic actions easily enough as follows: for a sequence $s = a_1 a_2 \cdots a_n \in \mathbf{Act}^*$, we say $p \xrightarrow{s} p'$ *iff* for some $p_0, p_1, \cdots, p_n$,

$$p = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{n-1}} p_{n-1} \xrightarrow{a_n} p_n = p'.$$

We shall often use this extension as a shorthand form of writing sequences of transitions.

Given any language $\mathcal{P}$ defined over a signature taken as a subset of that given in **Figure 2–1** but not including the recursion operator μ, any agent $p \in \mathcal{P}^0$ will have associated with it a *derivation tree*, a finite unordered tree whose nodes are labelled by terms of $\mathcal{P}^0$, and whose arcs are labelled by atomic actions from the

set **Act**. The root of the tree is labelled by the agent p itself, and each node has emitting from it an arc corresponding to each of the possible transitions which the agent can make, with the arc labelled by the action labelling the transition, and leading to a node labelled by the agent into which the source node agent would evolve under that transition. Thus the derivation tree of an agent specifies the possible transition sequences which a term can undergo. For example, the derivation tree of the agent $a \mid b + c$ is given in **Figure 2–3**.

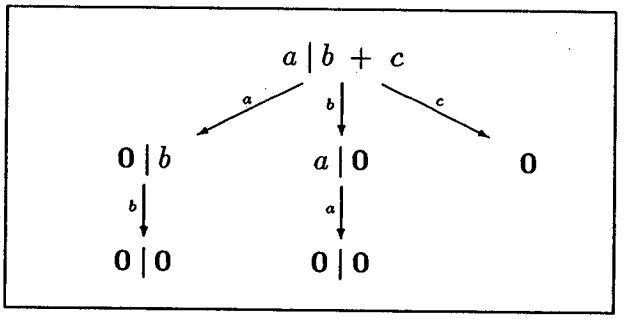

Figure 2–3: Derivation tree for $a \mid b + c$.

Motivated by the above definition of derivation trees, we can define some useful operations on agents, to specify the lengths of the longest and shortest possible transition sequences which an agent is capable of performing. Firstly, the former notion is given by the *depth* of the derivation tree, which is the longest path from the root of the tree to any of its leaves.

Definition 2.2.1 *The* depth *of a (finite) agent* $|\cdot|$ *is given by*

$$
\begin{aligned}
|0| &= 0; \\
|ap| &= 1 + |p|; \\
|p + q| &= \max\big(|p|, |q|\big); \\
|p \parallel q| &= |p| + |q|; \\
|p \lfloor\!\lfloor q| &= \begin{cases} |p| + |q|, & \text{if } |p| > 0, \\ 0, & \text{otherwise;} \end{cases} \\
|p \mid q| &= |p| + |q|.
\end{aligned}
$$

Secondly, the latter notion can be defined by defining the shortest path from the root of the tree to any of its leaves.

Definition 2.2.2 *The* shortest transition length *of a (finite) agent $\Omega(\cdot)$ is given by*

$$\begin{aligned}
\Omega(\mathbf{0}) &= 0;\\
\Omega(ap) &= 1+\Omega(p);\\
\Omega(p+q) &= \begin{cases} \Omega(p), & \text{if } |q|=0,\\ \Omega(q), & \text{if } |p|=0,\\ \min\big(\Omega(p),\Omega(q)\big), & \text{otherwise;}\end{cases}\\
\Omega(p \parallel q) &= \Omega(p)+\Omega(q);\\
\Omega(p \lfloor\!\lfloor q) &= \begin{cases} \Omega(p)+\Omega(q), & \text{if } |p|>0,\\ 0, & \text{otherwise;}\end{cases}\\
\Omega(p \mid q) &= \min\Big(\{0 \mid \nexists a,r \text{ st } p \xrightarrow{a} r \vee q \xrightarrow{a} r\}\\
&\qquad \cup \{1+\Omega(p' \mid q) \mid \exists a \text{ st } p \xrightarrow{a} p'\}\\
&\qquad \cup \{1+\Omega(p \mid q') \mid \exists a \text{ st } q \xrightarrow{a} q'\}\\
&\qquad \cup \{1+\Omega(p' \mid q') \mid \exists a \text{ st } p \xrightarrow{a} p' \wedge q \xrightarrow{\bar{a}} q'\}\Big).
\end{aligned}$$

We shall often want to extend these two measures to open terms, which we do as follows: for $\vec{x} = fv(t)$,

$$\begin{aligned}
|t| &= |t\{\overline{0}/\vec{x}\}|, \quad \textit{and}\\
\Omega(t) &= \Omega\left(t\{\overline{0}/\vec{x}\}\right).
\end{aligned}$$

With these definitions, we can state the following simple propositions.

Proposition 2.2.3 $|p| = \max\{n \mid \exists s \in \mathbf{Act}^n, \exists p' \text{ st } p \xrightarrow{s} p'\}$.

Proof:

By structural induction on p. □

Corollary 2.2.4 $p \xrightarrow{a} p' \implies |p| > |p'|$.

Proof:

$$|p'| = n \implies \exists s \in \mathbf{Act}^n, \exists p'' \; st \; p' \xrightarrow{s} p''$$
$$\implies p \xrightarrow{as} p''$$
$$\implies |p| > n = |p'|. \qquad \square$$

Proposition 2.2.5 $\Omega(p) = \min \{n \mid \exists s \in \mathbf{Act}^n, \exists p' \; st \; p \xrightarrow{s} p'$ *and* $\forall p'', \forall a \in \mathbf{Act}, \; p' \not\xrightarrow{a} p''\}$.

Proof:

By structural induction on p. $\square$

2.2.2 Strong Observational Congruence

The first and main congruence in which we shall be interested is known as *strong observational congruence*, or simply *strong congruence*. The basic idea behind this equivalence is as follows: given a process language $\mathcal{P}$, two agents taken from the sublanguage $\mathcal{P}^0$ of closed terms are deemed to be equivalent exactly when they share the same possibilities of acting (that is, they can perform exactly the same atomic actions according to the transition system defined as above), and can also evolve into equivalent processes upon doing identical actions. Formally, this notion is defined using the notion of a *strong bisimulation* relation (*cf.*, *e.g.*, [MIL80], or [PAR81]).

Definition 2.2.6 *A binary relation* $\mathcal{R} \subseteq \mathcal{P}^0 \times \mathcal{P}^0$ *is a* strong bisimulation *iff whenever* $p\mathcal{R}q$, *then for all* $a \in \mathbf{Act}$,

(*i*) $p \xrightarrow{a} p'$ *implies* $\exists q'$ *such that* $q \xrightarrow{a} q' \wedge p'\mathcal{R}q'$; *and*

(*ii*) $q \xrightarrow{a} q'$ *implies* $\exists p'$ *such that* $p \xrightarrow{a} p' \wedge p'\mathcal{R}q'$.

synchronisation, and should be unobservable to any observer of the system — the composite system is not communicating with the outside environment, but rather communicating internally. Thus we would like to abstract away from this internal silent or unobservable action τ.

The equivalence we shall define for this purpose is going to be a straightforward refinement of the strong observational equivalence defined above. However, instead of matching actions exactly, we shall only require observable actions to be matched, modulo the occurrence of any finite number of invisible τ actions before and after the observable action.

For this case, we make some assumptions about the action set **Act**, which are usual in the presentation of **CCS**. These are made to define what complementary actions are, to allow for two processes to synchronise or communicate. Firstly we split the silent action $\tau \in \mathbf{Act}$ away from the set $\Lambda \subseteq \mathbf{Act}$ of visible action, and express the set **Act** as $\mathbf{Act} = \Lambda \cup \{\tau\}$ (where $\tau \notin \Lambda$). Then we assume that we can further partition the set Λ of visible actions into two equinumerous disjoint sets $\Lambda = \Delta \cup \bar{\Delta}$, where $\bar{\Delta} = \{\bar{a} \mid a \in \Delta\}$. The mapping $a \mapsto \bar{a}$ defines a bijection from Δ to $\bar{\Delta}$, and is extended to all of Λ by defining $\bar{\bar{a}} = a$. Then two actions a and $\bar{a}$ are deemed to be *complementary* actions, and are the actions which can be used by two processes to synchronise, as specified by rule (5c) of **Figure 2–2.**

For $\mu \in \mathbf{Act}$ (either a visible action or τ), we shall say that $p \stackrel{\mu}{\Longrightarrow} p'$ *iff* $p \stackrel{\tau^m \mu \tau^n}{\longrightarrow} p'$ for some $m, n \geq 0$. We can again extend this definition to apply to sequences of atomic actions as follows: for a sequence $s = \mu_1 \mu_2 \cdots \mu_n \in \mathbf{Act}^*$, we say $p \stackrel{s}{\Longrightarrow} p'$ *iff* for some $p_0, p_1, \cdots, p_n$,

$$p = p_0 \stackrel{\mu_1}{\Longrightarrow} p_1 \stackrel{\mu_2}{\Longrightarrow} \cdots \stackrel{\mu_{n-1}}{\Longrightarrow} p_{n-1} \stackrel{\mu_n}{\Longrightarrow} p_n = p'.$$

We also allow for the case where $n = 0$ above in writing $p \stackrel{\epsilon}{\Longrightarrow} p'$ (or also written simply as $p \Longrightarrow p'$) whenever $p \stackrel{\tau^m}{\Longrightarrow} p'$ for some $m \geq 0$. With this new transition system, we can now define our refined equivalence which will abstract away from internal silent τ actions. This we do using the notion of a *weak bisimulation* relation (*cf.*, *e.g.*, [MIL80], [MIL85]).

Definition 2.2.8 *A relation $\mathcal{R} \subseteq \mathcal{P}^0 \times \mathcal{P}^0$ is a* weak bisimulation *iff whenever $p\mathcal{R}q$, then for all $s \in \Lambda^*$,*

$$(i)\ p \stackrel{s}{\Longrightarrow} p' \quad \textit{implies} \quad \exists q' \textit{ such that } q \stackrel{s}{\Longrightarrow} q' \wedge p'\mathcal{R}q'\textit{; and}$$
$$(ii)\ q \stackrel{s}{\Longrightarrow} q' \quad \textit{implies} \quad \exists p' \textit{ such that } p \stackrel{s}{\Longrightarrow} p' \wedge p'\mathcal{R}q'.$$

Then with this definition, we say that two agents $p, q \in \mathcal{P}^0$ are *weakly observationally equivalent* (or simply *observationally equivalent*), or *weakly bisimilar*, written $p \approx q$, iff there is a weak bisimulation $\mathcal{R}$ containing the pair (p, q). Again the relation $\approx$ defined here is the *largest* weak bisimulation, and is easily seen to be an equivalence relation. However, it is well known *not* to be a congruence relation. This is revealed by the following counterexample: we can easily show that $a \approx \tau a$, but $a + aa \not\approx \tau a + aa$ (as $\tau a + aa \stackrel{\epsilon}{\Longrightarrow} a$, but for no $p \approx a$ does $a + aa \stackrel{\epsilon}{\Longrightarrow} p$). We would dearly like our equivalence to be a congruence relation though — if two process terms are to be deemed equivalent, we would like them to be interchangeable as subterms in some bigger expression. That is, given any *context* $C[\cdot]$ (*i.e.*, a term with a *"hole"* in it), we would like to be assured that whenever $p \approx q$ we have that $C[p] \approx C[q]$. Such a property would assure us that we could safely interchange equivalent programs as subprograms of a larger system. Thus we shall refine our equivalence by defining $\approx^c$ to be the largest congruence contained in $\approx$. This congruence is known as *(weak) observational congruence*, and gives us exactly the relation we are looking for: $p \approx^c q$ *iff* for all contexts $C[\cdot]$, $C[p] \approx C[q]$.

We can actually define this congruence directly using the following proposition.

Proposition 2.2.9 *$p \approx^c q$ iff for all $\mu \in \mathbf{Act}$,*

$$(i)\ p \stackrel{\mu}{\longrightarrow} p' \quad \textit{implies} \quad \exists q' \textit{ such that } q \stackrel{\mu}{\Longrightarrow} q' \wedge p' \approx q'\textit{; and}$$
$$(ii)\ q \stackrel{\mu}{\longrightarrow} q' \quad \textit{implies} \quad \exists p' \textit{ such that } p \stackrel{\mu}{\Longrightarrow} p' \wedge p' \approx q'.$$

Proof:

See e.g., [MIL85], **Proposition 2.6.** □

We shall freely use this alternate characterisation when dealing with observational congruence in the sequel.

One last concept which we shall find useful when dealing with action sequences involving silent τ actions is as follows.

Definition 2.2.10 *The* τ-free projection $\widehat{s}$ *of a string* $s \in \mathbf{Act}^*$ *is defined to be the string* s *with all occurrences of* τ *removed. Formally, we have the following:*

$$\begin{aligned} \widehat{a} &= a; \quad (a \in \Lambda) \\ \widehat{\tau} &= \epsilon; \\ \widehat{s_1 \cdot s_2} &= \widehat{s_1} \cdot \widehat{s_2}. \end{aligned}$$

2.3 Equational Characterisations

Once we have defined a particular observational congruence as above using the notion of bisimulations, we would like to have some method of determining when two processes are equivalent. The general technique is to try to construct a bisimulation relating the two processes in question. Techniques for doing this can be found in *e.g.,* [MIL80] and [SAN82]. However as we have pointed out, it is not always sufficient simply to show that two processes are related in some bisimulation, as the congruence we are actually interested in is not always identical to the equivalence defined by the bisimulation notion. Also, constructing bisimulations (and proving that the relation constructed *is* a bisimulation) is often not such a straightforward task. Fortunately, much effort has been expended on deriving alternate characterisations of observational congruences. In particular, equational axiomatisations for different process algebras abound. Quite clearly, equational proofs are bound to be much simpler than bisimulation constructions. Furthermore equational systems, involving the laws of equivalences (reflexivity, symmetry and transitivity) *as well as* the law of substitutivity, naturally define congruences. Thus we have no problems of mismatch between our semantic congruence and the notion defining the basis of our proof technique.

2.3.1 An Equational Proof System

Given a process language defined over a signature, and a set of equational axioms over that language, we can generate equivalences between terms in the language by using axioms or applying the rules of equational logic: reflexivity, symmetry, transitivity and substitutivity. There are several different equivalent approaches for defining a formal system for doing these syntactic manipulations. As we would like to prove some results on the limitations of proving statements in our algebras using equational logic, we would like now to fix a particular system about which to argue. Thus in this section we shall present a *natural deduction style* proof system following [PRA65] parameterised by a set $\mathcal{T}$ of equational axioms.

Our proof system will allow proofs of equivalences in the forms of *proof trees*, where the trees are put together by *inferences*. The inferences will be of the following form:

$$\frac{\cdots,\ t_i = u_i,\ \cdots}{t = u}(rule).$$

This inference is meant to state that from the set of premises $\{\cdots,\ t_i = u_i,\ \cdots\}$ we can assume the conclusion $t = u$. A valid *proof* of a statement will then be a *finite* proof tree built up from such inferences, where the statement being proven is the lone statement at the root of the tree (at the bottom), and there are no premises at the leaves of the tree (that is, all topmost inferences have empty sets of premises).

The inferences which are allowed are as expected: firstly, every axiom of $\mathcal{T}$ can be instantiated with an empty premise:

$$\frac{}{t\sigma = u\sigma}(t = u) \qquad \textit{(where } t = u \in \mathcal{T}\textit{, and } \sigma \textit{ is some substitution).}$$

An axiom is intended to state a universally true fact. Then we need to allow inferences based on the laws of equational reasoning. The first three, corresponding to reflexivity, symmetry and transitivity, are as follows:

$$\frac{}{t = t}(refl), \qquad \frac{t = u}{u = t}(symm), \qquad \frac{t = u,\ \ u = v}{t = v}(trans).$$

The final set of inferences correspond to substitutivity, and are taken from the subset of the following inferences involving the operators in the signature of the language which we happen to be considering.

$$\frac{t = u}{\alpha t = \alpha u}(sub_\alpha)$$

$$\frac{t_1 = u_1,\ t_2 = u_2}{t_1 + t_2 = u_1 + u_2}(sub_+) \qquad \frac{t_1 = u_1,\ t_2 = u_2}{t_1 \parallel t_2 = u_1 \parallel u_2}(sub_\parallel)$$

$$\frac{t_1 = u_1,\ t_2 = u_2}{t_1 \between t_2 = u_1 \between u_2}(sub_\between) \qquad \frac{t_1 = u_1,\ t_2 = u_2}{t_1 \mid t_2 = u_1 \mid u_2}(sub_\mid)$$

$$\frac{t = u}{\mu x.t = \mu x.u}(sub_\mu)$$

Thus for a given set of axioms $\mathcal{T}$, our fixed proof system will be the above natural deduction style system. Whenever we can produce a proof of a statement $t = u$ in this system, we shall denote this fact either by $\mathcal{T} \vdash t = u$ or equivalently by $t =_{\mathcal{T}} u$. Having fixed a firm formal system, we can do rigorous proofs on certain properties of equational logic for our process algebras. For instance, this formalism will figure prominently in our proofs of the nonexistence of finite equational axiomatisations in what follows.

We shall sometimes want to discuss a certain extension to the above proof system, and allow not just ordinary axioms in the set $\mathcal{T}$, but also include *conditional axioms* of the form

$$\frac{t_i = u_i \quad (1 \leq i \leq n)}{t = u}.$$

These new inferences fit well into our framework, and will be included as valid inferences in a proof tree whenever they appear in the set $\mathcal{T}$ of axioms.

2.3.2 Soundness, Completeness, and ω-Completeness

The purpose of defining our proof system above was so that we could use equational logic to syntactically reason about processes instead of having to give semantic

justifications for relating processes. Thus if we intend on using our equational system, we would like to be sure that any equivalences which we can possibly generate are going to be valid semantic equivalences, and that if two agents are semantically equivalent, then we can prove them to be so in our syntactic formal system. These two important notions are respectively referred to as *soundness* and *completeness* of a system, and are formally defined as follows.

1. **(Soundness:)** For a given congruence $\cong$ over a process language $\mathcal{P}$, an axiom system $\mathcal{T}$ is sound *iff* for all terms $t, u \in \mathcal{P}$ we have that $\mathcal{T} \vdash t = u \implies t \cong u$.

2. **(Completeness:)** For a given congruence $\cong$ over a process language $\mathcal{P}$, an axiom system $\mathcal{T}$ is complete *iff* for all *(closed)* terms $p, q \in \mathcal{P}^0$ we have that $p \cong q \implies \mathcal{T} \vdash p = q$.

Equational systems satisfying the above for different process algebras are abundant, and many shall be referred to in the following chapters of this thesis. However, one of the motivations of this thesis is the axiomatisability of the theory of open term reasoning. In the above definition of completeness, we are only guaranteed to prove valid equivalences between closed terms in our language. We shall indeed see that in general there are valid equations relating open terms which are not derivable in a system which is *"complete"* by the above definition. Thus we could not rely just on equational logic to prove valid equivalences between underdefined processes (those expressed by terms containing free process variables). In general, we would need to invoke extra techniques (such as structural induction) to prove such equivalences.

This deficiency leads us to define the following stronger notion of completeness, *ω-completeness*, which captures more closely the dual property of soundness.

3. **(ω-Completeness:)** For a given congruence $\cong$ over a process language $\mathcal{P}$, an axiom system $\mathcal{T}$ is ω-complete *iff* for all terms $t, u \in \mathcal{P}$ we have that $t \cong u \implies \mathcal{T} \vdash t = u$.

The term and the concept of ω-completeness are used in [HEE86] in studying algebraic specifications. However, this definition originally arises from the definition of the infinitary *ω-rule* of the λ-calculus (*cf.*, *e.g.*, [BAR84], [HIN86]):

$$\frac{MZ = NZ \quad \forall Z}{M = N}(\omega\text{-}rule).$$

We could get the power of this definition in our equational system easily enough by simply allowing such a conditional axiom into our system; that is, allowing into our axiom set the following law:

$$\frac{t\sigma = u\sigma \quad \forall \sigma}{t = u}$$

However, we do not wish to allow this for two reasons. Firstly, we shall usually be concerned with systems parameterised by a set $\mathcal{T}$ of unconditional axioms, so we would not want to allow such a conditional law. More importantly though, we wish to stay completely within equational reasoning. In order to use such an infinitary law in a finite proof, we would need to invoke some extra powerful proof strategy such as some form of induction just to generate the infinite set of premises.

Chapter 3

ω-Complete Axiomatisations

Complete equational axiomatisations for process algebras exist in abundance. For example, almost any standard reference on **CCS** (*e.g.,* [MIL80], [MIL85]), **CSP** (*e.g.,* [HOA85], [BRO84]), or **ACP** (*e.g.,* [BER84], [BER85]) will list several axiomatisations for different languages, along with proofs of soundness and completeness of the axiomatisations with respect to some semantic congruence. Indeed, we shall meet several such axiomatisations for algebras based on **CCS** in this chapter.

However, rarely do these references deal with anything but closed-term reasoning, with the exception of [MIL84] and [MIL86], which (necessarily) deal with the problem of relating open terms over the language of *regular (finite-state) behaviours.* In these latter studies, the equational theories for closed-term and open-term reasoning coincide, so that the theory of closed-term reasoning is in itself sufficiently powerful to prove true any valid open-term statement. Such is the simplicity of the algebra that the ω-completeness of the axiom set comes along with the proof of simple completeness. However, this is not in general the case when considering more complicated process algebras. Often it is the case that, though a complete set of laws for some congruence over a process algebra is certainly sound for open-term reasoning, it need not be sufficient to prove all possible valid open statements.

In this chapter, we investigate the axiomatisations of open theories for various subsets of **CCS**. We shall see the mismatch occurring in the closed-term and open-term equational theories, and discover that though it is often a simple matter

deriving a *complete* set of axioms for a given process algebra, this is by no means the case when looking for an *ω-complete* set of axioms.

3.1 A Simple Nondeterministic Language

We begin our study of ω-complete axiomatisations with a consideration of a simple algebra containing no operators for concurrent computation. The results presented in this section are precisely a subset of those presented in [MIL84]. However, we prove our results here without Milner's (implicit) assumption that we have an infinite (non-exhaustive) action set[1].

3.1.1 Finite Terms

The first language $\mathcal{P}_0$ which we shall consider is a simple language of finite nondeterministic terms given by the signature $\Sigma_0 = \{\mathbf{0}, ., +\}$. The semantic equivalence which we consider here will be the strong observational congruence $\sim$ defined in **Section 2.2.2.**

The equational theory which we shall prove to be identical to the semantic equivalence is the theory $\mathcal{T}_0$ consisting of the following four axioms:

$$(A_1)\ \ (x+y)+z = x+(y+z); \qquad (A_3)\ \ x+x = x;$$
$$(A_2)\ \ x+y = y+x; \qquad (A_4)\ \ x+\mathbf{0} = x.$$

The first point to notice about these axioms is that they characterise the usual theory of strong observational congruence between closed terms, as shown in the following proposition:

Proposition 3.1.1 *The strong observational congruence over $\mathcal{P}_0^0$, the subset of closed terms of the language $\mathcal{P}_0$ is exactly the congruence induced by the four axioms of $\mathcal{T}_0$ given above.*

---

[1]See the remark in **Section 5.2** of [MIL80]

Proof:

See, e.g., [HEN85], **Theorem 3.1**. *Alternately, the result follows from the proofs in this section of the soundness and completeness of $\mathcal{T}_0$ for arbitrary open terms (and thus for closed terms as well).* □

Thus it turns out that we need not add any new axioms to deal with open terms of the language. The proof of the fact that these four axioms exactly characterise the congruence in the open theory is similar to the proof given in [HEN85] for the above **Proposition 3.1.1**. It is broken down into two parts, proving soundness and completeness of the axioms separately.

Proposition 3.1.2 (Soundness) $\mathcal{T}_0 \vdash t = u \implies t \sim u$.

Proof:

It simply requires to show that for all terms t, u, and v:

1. $(t+u)+v \sim t+(u+v)$;
2. $t+u \sim u+t$;
3. $t+t \sim t$;
4. $t+\mathbf{0} \sim t$.

But all ground instances of these hold (in each case, denoting left and right sides by p and q, one can show from the definition of $\xrightarrow{a}$ that $p \xrightarrow{a} p'$ iff $q \xrightarrow{a} p'$), so the four laws follow immediately from the ω-completeness rule for the semantic equivalence of open terms. □

The proof of completeness does not come so quickly, and will be treated in depth, to set the stage for the more complicated languages which we shall be considering later. The proof comes out of a sequence of propositions which define and manipulate normal forms for terms of the language. The normal forms are defined using a *denotation* function which distinguishes between non-equivalent process terms. The domain of values to which the denotation maps terms is given by the least fixed point solution $\mathcal{D}_0$ to the set equation

$$\mathcal{D}_0 = \mathcal{P}_{FIN}\big(\mathbf{Var} \cup \mathbf{Act} \times \mathcal{D}_0\big),$$

where $\mathcal{P}_{FIN}(S)$ represents the set of *finite* subsets of S.

Definition 3.1.3 *The* denotation *of terms* $[\![\cdot]\!] : \mathcal{P}_0 \to \mathcal{D}_0$ *is given by:*

$$[\![0]\!] = \emptyset \qquad [\![at]\!] = \{(a, [\![t]\!])\}$$
$$[\![x]\!] = \{x\} \qquad [\![t + u]\!] = [\![t]\!] \cup [\![u]\!]$$

Thus informally, the denotation of a term is a set containing all *unguarded* occurrences of variables in the term (those variables which appear but not as a subterm in an action-prefixed term), as well as tuples representing the immediate actions which the term can perform, together with the denotations of the resulting terms into which the original term evolves. Being a set, idempotence is accounted for (capturing axiom (A_3)), the $\mathbf{0}$ process is absorbed (capturing axiom (A_4)), and order is ignored between summands (capturing axioms (A_1) and (A_2)).

The important properties which we shall use about the denotations of terms are the following.

Proposition 3.1.4 $(a,T) \in [\![t]\!]$ *iff* $\exists t'$ *such that* $[\![t']\!] = T$ *with* $t \xrightarrow{a} t'$.

Proof:

By structural induction on t. □

Proposition 3.1.5 $t\{\bar{p}/\bar{x}\} \xrightarrow{a} p$ *iff either*

(i) $\exists t' : t \xrightarrow{a} t'$ *such that* $p = t'\{\bar{p}/\bar{x}\}$; *or*

(ii) $\exists x \in [\![t]\!]$ *such that* $p_x \xrightarrow{a} p$.

Proof:

By structural induction on t. □

This proposition easily generalises to a sequence of actions as follows.

Corollary 3.1.6 $t\{\bar{p}/\bar{x}\} \xrightarrow{s} p$ *iff either*

(i) $\exists t' : t \xrightarrow{s} t'$ *such that* $p = t'\{\bar{p}/\bar{x}\}$; *or*

(ii) $\exists s_1, s_2, t', x$ *with* $s = s_1 s_2$ *such that* $t \xrightarrow{s_1} t'$, $x \in [\![t']\!]$, *and* $p_x \xrightarrow{s_2} p$.

Proof:

By induction on the length of s. □

The denotation function is used to define the *normal form* of a term, which will be an equivalent term which is expressed as a sum of action-prefixed normal form terms added to a sum of variables. The normal form of a term is extracted from its denotation as follows.

Definition 3.1.7 *The* normal form *of terms* $nf(\cdot)$ *is given by:*

$$nf(t) = \sigma([\![t]\!]),$$

where

$$\sigma(T) = \sum_{(a,S)\in T} a.\sigma(S) + \sum_{x\in T} x.$$

By convention, we let $\sigma(\emptyset) = \mathbf{0}$.

Proposition 3.1.8 $\mathcal{T}_0 \vdash t = nf(t)$.

Proof:

By structural induction on t.

- $\mathbf{0} = \sigma(\emptyset) = \sigma([\![\mathbf{0}]\!]) = nf(\mathbf{0})$;
- $x = \sigma(\{x\}) = \sigma([\![x]\!]) = nf(x)$;

- $\mathcal{T}_0 \vdash t = nf(t) = \sigma([\![t]\!])$
$$\Longrightarrow at =_{\mathcal{T}_0} a(\sigma([\![t]\!])) = \sigma(\{(a, [\![t]\!])\}) = \sigma([\![at]\!]) = nf(at);$$

- $\mathcal{T}_0 \vdash t = nf(t) = \sigma([\![t]\!])$ *and* $\mathcal{T}_0 \vdash u = nf(u) = \sigma([\![u]\!])$
$$\Longrightarrow t + u =_{\mathcal{T}_0} \sigma([\![t]\!]) + \sigma([\![u]\!]) =_{\mathcal{T}_0} \sigma([\![t]\!] \cup [\![u]\!]) = \sigma([\![t + u]\!]) = nf(t + u). \qquad \square$$

Corollary 3.1.9 $[\![t]\!] = [\![u]\!] \Longrightarrow \mathcal{T}_0 \vdash t = u.$

Proof:

$$t =_{\mathcal{T}_0} nf(t) = \sigma([\![t]\!]) = \sigma([\![u]\!]) = nf(u) =_{\mathcal{T}_0} u. \qquad \square$$

The next proposition is the main part of our completeness proof. It shows that if two terms have distinct denotations, then there will be instantiations for the variables in the terms which give rise to non-observationally congruent terms. Hence the two terms will themselves be noncongruent according to our extensional definition for observational congruence. The interesting point about the proposition is that it places no unnecessary restrictions on the action set **Act**; it merely assumes that the set is nonempty – that there exists some $a \in \mathbf{Act}$. Clearly if this were not the case, then there would be no observable difference between any agents, and any observational congruence would collapse into the trivial congruence equating all terms.

Proposition 3.1.10 *Suppose that* $[\![t]\!] \neq [\![u]\!]$*; let* $fv(t) \cup fv(u) \subseteq \bar{x} = \{x_1, \ldots, x_n\}$, $m > \max(|t|, |u|)$, $a \in \mathbf{Act}$, *and* $\bar{p} = \{\mathcal{A}_m, \mathcal{A}_{2m}, \ldots, \mathcal{A}_{nm}\}$, *where* $\mathcal{A}_0 = \mathbf{0}$ *and* $\mathcal{A}_{k+1} = a\mathcal{A}_k$*; then* $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$.

Proof:

By induction on $|t| + |u|$. *Suppose* $[\![t]\!] \neq [\![u]\!]$, *and let* $m, \bar{x}, \bar{p}$ *be given as above; there are four cases to consider:* $x \in [\![t]\!] \setminus [\![u]\!]$, $x \in [\![u]\!] \setminus [\![t]\!]$, $(b,T) \in [\![t]\!] \setminus [\![u]\!]$, $(b,U) \in [\![u]\!] \setminus [\![t]\!]$.

- *Suppose* $x_k \in [\![t]\!]$, *but* $x_k \notin [\![u]\!]$;

  then by **Corollary 3.1.6**, $t\{\bar{p}/\bar{x}\} \xrightarrow{a^{km}} \mathbf{0}$;
  thus $t\{\bar{p}/\bar{x}\} \sim u\{\bar{p}/\bar{x}\} \implies \exists p \sim \mathbf{0}$ *st* $u\{\bar{p}/\bar{x}\} \xrightarrow{a^{km}} p$;
  however, if $u\{\bar{p}/\bar{x}\} \xrightarrow{a^{km}} p \sim \mathbf{0}$,
  then again from **Corollary 3.1.6**, *either:*
  (i) $\exists u' : u \xrightarrow{a^{km}} u'$ *st* $p = u'\{\bar{p}/\bar{x}\}$ *(which is impossible, as* $|u| < m \leq km$, *so by* **Proposition 2.2.3**, $u \not\xrightarrow{a^{km}}$*);*
  or
  (ii) $\exists i : 0 \leq i \leq |u| < m$, u', j *st* $u \xrightarrow{a^i} u'$ *with* $x_j \in [\![u']\!]$ *st* $p_j \xrightarrow{a^{km-i}} p$;
  but $p_j \xrightarrow{a^{km-i}} p \sim \mathbf{0}$
  $\implies i = (k-j)m$
  $\implies j = k$, $i = 0$, *and* $u' = u$ *(as* $i < m$*);*
  but then $x_k \in [\![u]\!] = [\![u']\!]$ *(contradiction);*
  thus $u\{\bar{p}/\bar{x}\} \not\xrightarrow{a^{km}} p \sim \mathbf{0}$, *so* $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Similarly,* $x_k \in [\![u]\!]$ *and* $x_k \notin [\![t]\!] \implies t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Suppose* $(b,T) \in [\![t]\!]$, *but* $(b,T) \notin [\![u]\!]$;

  by **Proposition 3.1.4**, $\exists t'$ *st* $[\![t']\!] = T$ *with* $t \xrightarrow{b} t'$;
  so by **Proposition 3.1.5**, $t\{\bar{p}/\bar{x}\} \xrightarrow{b} t'\{\bar{p}/\bar{x}\}$;
  thus $t\{\bar{p}/\bar{x}\} \sim u\{\bar{p}/\bar{x}\}$
  $\implies \exists q \sim t'\{\bar{p}/\bar{x}\}$ *st* $u\{\bar{p}/\bar{x}\} \xrightarrow{b} q$;
  hence by **Proposition 3.1.5**, *either:*
  (i) $\exists u' : u \xrightarrow{b} u'$ *st* $q = u'\{\bar{p}/\bar{x}\} \sim t'\{\bar{p}/\bar{x}\}$;
  thus by **Proposition 3.1.4**, $(b, [\![u']\!]) \in [\![u]\!]$,
  so since $(b,T) \notin [\![u]\!]$, $[\![u']\!] \neq T = [\![t']\!]$;
  hence by the induction hypothesis (as from **Proposition 2.2.4**, $|t'| + |u'| < |t| + |u|$*),*

$$t'\{\bar{p}/\bar{x}\} \not\sim u'\{\bar{p}/\bar{x}\} \text{ (contradiction);}$$

or

(ii) $\exists x \in [\![u]\!]$ *st* $p_x \xrightarrow{b} q$;

that is, $b = a$, *and* $\exists k > 0$ *st*

$$x_k \in [\![u]\!] \text{ and } q = \mathcal{A}_{km-1};$$

hence $q \xrightarrow{a^{km-1}} \mathbf{0}$, *so* $\exists p \sim \mathbf{0}$ *st* $t'\{\bar{p}/\bar{x}\} \xrightarrow{a^{km-1}} p$;

thus by **Corollary 3.1.6**, *either:*

(a) $\exists t'' : t' \xrightarrow{a^{km-1}} t''$ *st* $p = t''\{\bar{p}/\bar{x}\} \sim \mathbf{0}$ *(which is impossible, as* $|t'| \leq |t| - 1 < m - 1 \leq km - 1$, *so* $t' \not\xrightarrow{a^{km-1}}$*);*

or

(b) $\exists i : 0 \leq i \leq |t'| < m - 1$, t'', j *st* $t' \xrightarrow{a^i} t''$ *with* $x_j \in [\![t'']\!]$ *st* $p_j \xrightarrow{a^{km-1-i}} p \sim \mathbf{0}$;
but $p_j \xrightarrow{a^{km-1-i}} p \sim \mathbf{0}$
$\implies i = (k-j)m - 1 \geq 0$;
(contradiction, as $i < m - 1$*)*

thus $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Similarly,* $(b, U) \in [\![u]\!]$ *and* $(b, U) \notin [\![t]\!] \implies t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$.

□

Corollary 3.1.11 $t \sim u \implies [\![t]\!] = [\![u]\!]$.

Proof:

Suppose $[\![t]\!] \neq [\![u]\!]$; *let* $\bar{x}$, $\bar{p}$ *be as in* **Proposition 3.1.10**;
then $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$; *hence by the ω-completeness definition of the semantic equivalence of open terms,* $t \not\sim u$. □

Corollary 3.1.12 (ω-Completeness) $t \sim u \implies \mathcal{T}_0 \vdash t = u$.

Proof:

Follows from **Corollary 3.1.11** *and* **Corollary 3.1.9.** □

As one last note to make about this sublanguage of nondeterministic terms, the results presented here are basically as presented in [MIL84] and [MIL86] with respect to *regular behaviours*. There Milner dealt with open terms of his language in a different but equivalent manner. He firstly defined what it meant for a variable to appear unguarded in a term as follows.

Definition 3.1.13 *The set of* unguarded variables *of a term,* $\mathcal{U}(\cdot)$*, is given by:*

$$\mathcal{U}(\mathbf{0}) = \emptyset \qquad \mathcal{U}(at) = \emptyset$$
$$\mathcal{U}(x) = \{x\} \qquad \mathcal{U}(t+u) = \mathcal{U}(t) \cup \mathcal{U}(u)$$

He then incorporated the effect of unguarded variables on terms into his definition of bisimulation as follows.

Definition 3.1.14 $t \cong u$ *iff for all* $a \in$ **Act***:*

(i) $t \xrightarrow{a} t'$ *implies* $\exists u'$ *such that* $u \xrightarrow{a} u'$ *and* $t' \cong u'$*;*

(ii) $u \xrightarrow{a} u'$ *implies* $\exists t'$ *such that* $t \xrightarrow{a} t'$ *and* $t' \cong u'$*;*

(iii) $\mathcal{U}(t) = \mathcal{U}(u)$.

Then the proposition which he states in [MIL86] without proof (as we do here) is as follows:

Proposition 3.1.15 $t \cong u$ *iff* $t \sim u$.

Using similar notions as found in this section, **Proposition 3.1.15** can be proven assuming only a singleton action set. However, as we pointed out at the start of this section, Milner's idea with his statement of the proposition was to allow a potentially infinite action set.

3.1.2 Regular Behaviours

The regular behaviours defined in [MIL84] and [MIL86] also allow for recursive definitions of terms, and **Proposition 3.1.15** above relating $\cong$ and $\sim$ still holds valid (where our definitions are suitably extended to recursive terms where necessary). However in order to prove this under the assumption of a singleton action set, we must utilise a more clever trick than that employed in the previous section. We must encode not the *depths* of the terms in the processes replacing the variables, but rather the *number of states* of the processes; being regular behaviours, all of the terms represent finite-state agents, so finite terms all of the form $\mathcal{A}_m$ as above can be instantiated for the variables of non-congruent terms to get closed instances of the processes which are not congruent.

3.2 Concurrency With The Left Merge Operator

The next language $\mathcal{P}_1$ which we shall consider is a language which incorporates nondeterministic choice as before, along with two operators for concurrent computation, the full merge operator $\|$, and the left merge operator $\lfloor\!\lfloor$, introduced by Hennessy and analysed extensively by Bergstra and Klop *et al.* The language is thus defined by the signature $\Sigma_1 = \{\mathbf{0}, ., +, \lfloor\!\lfloor\}$.

Notice here that our signature does not include the full merge operator $\|$. This is because in fact as defined operationally in **Figure 2–1**, we need not introduce the full merge operator into our signature at all, as we could define it simply enough in terms of the left merge and choice operators as follows.

$$t \| u \stackrel{\text{def}}{=} t \lfloor\!\lfloor u + u \lfloor\!\lfloor t.$$

Defined in this way, the operator behaves operationally precisely as expected: the actions of the two process terms represented by its operands are interleaved nondeterministically, with the choice of whether the first action deriving from the

first operand or the second operand itself being made nondeterministically. Thus we shall only treat this operator as a form of *syntactic sugar*, allowing us to all but ignore its existence in our analysis, forever reading $t \parallel u$ as $t \lfloor\!\lfloor u + u \lfloor\!\lfloor t$.

The semantic equivalence which we consider here will again be the strong observational congruence $\sim$ defined above. The equational theory which we shall prove to be identical to the semantic equivalence is the theory $\mathcal{T}_1$ consisting of the following nine axioms.

(A_1)	$(x+y)+z = x+(y+z)$	(L_1)	$\mathbf{0} \lfloor\!\lfloor x = \mathbf{0}$
(A_2)	$x+y = y+x$	(L_2)	$\alpha x \lfloor\!\lfloor y = \alpha(x \parallel y)$
(A_3)	$x+x = x$	(L_3)	$(x+y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$
(A_4)	$x+\mathbf{0} = x$	(L_4)	$(x \lfloor\!\lfloor y) \lfloor\!\lfloor z = x \lfloor\!\lfloor (y \parallel z)$
		(L_5)	$x \lfloor\!\lfloor \mathbf{0} = x$

If our signature had included the full merge operator $\parallel$, then we would simply need to add to the above list of axioms the defining law $x \parallel y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x$.

The first thing to notice here is that we now have a richer axiom system than that usually defined for dealing with closed expressions of this language. The final axioms (L_4) and (L_5) can be omitted — which is what is always done — when considering closed terms, as all closed instances of these axioms can be proven by structural induction using the other seven axioms. Thus we have the following result.

Proposition 3.2.1 *The observational congruence over $\mathcal{P}_1^0$, the sublanguage of closed terms of the language $\mathcal{P}_1$, is exactly characterised by the theory $\mathcal{T}_1^0$ consisting of the eight axioms $(A_1)-(A_4)$, $(L_1)-(L_4)$ given above.*

Proof:

All closed instances of (L_5) can be proven by structural induction using the axioms of $\mathcal{T}_1^0$; so combining this with the results to follow on the soundness and completeness of $\mathcal{T}_1$ for arbitrary open terms (and thus closed terms as well), we arrive at our result. □

However, the system $\mathcal{T}_1^0$ itself is not enough to completely reason about *open* terms over $\mathcal{P}_1$. In particular, we shall show that $\mathcal{T}_1^0 \not\vdash (L_5)$, which by the **Soundness Proposition 3.2.4** below is a valid law.

Proposition 3.2.2 *Let:*

$$\begin{aligned} V(\mathbf{0}) &= \mathit{false}; & F(\mathbf{0}) &= \mathit{false};\\ V(x) &= \mathit{true}; & F(x) &= \mathit{false};\\ V(at) &= \mathit{false}; & F(at) &= \mathit{false};\\ V(t+u) &= V(t) \vee V(u); & F(t+u) &= F(t) \vee F(u);\\ V(t \between u) &= V(t); & F(t \between u) &= V(t). \end{aligned}$$

(Thus intuitively, $V(t) = true$ iff the first action of t can be taken from a variable process, and $F(t) = true$ iff the first action of t forced by a "$\between$" can be taken from a variable). Then

$$\mathcal{T}_1^0 \vdash t = u \implies F(t) = F(u) \wedge V(t) = V(u).$$

Proof:

We need to show that this property is preserved by reflexivity, symmetry, transitivity, and substitutivity, as well as by all the axioms of $\mathcal{T}_1^0$:

$$\underline{(\mathit{refl})}: \frac{}{F(t) = F(t) \wedge V(t) = V(t)};$$

$$\underline{(\mathit{symm})}: \frac{F(t) = F(u) \wedge V(t) = V(u)}{F(u) = F(t) \wedge V(u) = V(t)};$$

$$\underline{(\mathit{trans})}: \frac{\begin{array}{c}F(t) = F(u) \wedge V(t) = V(u),\\ F(u) = F(v) \wedge V(u) = V(v)\end{array}}{F(t) = F(v) \wedge V(t) = V(v)};$$

$$\underline{(\mathit{sub}_\alpha)}: \frac{F(t) = F(u) \wedge V(t) = V(u)}{F(at) = F(au) \wedge V(at) = V(au)};$$

$$\underline{(sub_+)}: \frac{\begin{array}{c}F(t_0)=F(u_0)\ \wedge\ V(t_0)=V(u_0),\\ F(t_1)=F(u_1)\ \wedge\ V(t_1)=V(u_1)\end{array}}{F(t_0+t_1)=F(u_0+u_1)\wedge V(t_0+t_1)=V(u_0+u_1)};$$

$$\underline{(sub_{\|})}: \frac{\begin{array}{c}F(t_0)=F(u_0)\ \wedge\ V(t_0)=V(u_0),\\ F(t_1)=F(u_1)\ \wedge\ V(t_1)=V(u_1)\end{array}}{F(t_0 \| t_1)=F(u_0 \| u_1)\ \wedge\ V(t_0 \| t_1)=V(u_0 \| u_1)};$$

$$\underline{(A1)}: F((t+u)+v)=F(t+(u+v)) \wedge V((t+u)+v)=V(t+(u+v));$$

$$\underline{(A2)}: F(t+u)=F(u+t)\ \wedge\ V(t+u)=V(u+t);$$

$$\underline{(A3)}: F(t+t)=F(t)\ \wedge\ V(t+t)=V(t);$$

$$\underline{(A4)}: F(t+\mathbf{0})=F(t)\ \wedge\ V(t+\mathbf{0})=V(t);$$

$$\underline{(L1)}: F(\mathbf{0} \| t)=F(\mathbf{0})\ \wedge\ V(\mathbf{0} \| t)=V(\mathbf{0});$$

$$\underline{(L2)}: F(at \| u)=F(a(t \| u))\ \wedge\ V(at \| u)=V(a(t \| u));$$

$$\underline{(L3)}: F((t+u) \| v)=F(t \| v+u \| v) \wedge V((t+u) \| v)=V(t \| v+u \| v);$$

$$\underline{(L4)}: F((t \| u) \| v)=F(t \| (u \| v)) \wedge V((t \| u) \| v)=V(t \| (u \| v)).$$

These are all easily seen to hold. □

Corollary 3.2.3 $T_1^0 \not\vdash x \| \mathbf{0} = x$.

Proof:

$F(x \| \mathbf{0}) = \mathit{true} \neq \mathit{false} = F(x)$;

Hence by **Proposition 3.2.2**, $T_1^0 \not\vdash x \| \mathbf{0} = x$. □

We may now continue on to demonstrate that the theory $\mathcal{T}_1$ completely characterises the extensional strong observational congruence over the language $\mathcal{P}_1$. Again we shall break our proof into two sections, a proof of *soundness* and a proof of *completeness* of the axioms of $\mathcal{T}_1$. Firstly, the easy half is soundness:

Proposition 3.2.4 (Soundness) $\mathcal{T}_1 \vdash t = u \implies t \sim u$.

Proof:

We only need to show that for all ground terms p, q, and r:

1.	$(p+q)+r$	$\sim$	$p+(q+r)$	5.	$\mathbf{0}\lfloor p$	$\sim$	$\mathbf{0}$
2.	$p+q$	$\sim$	$q+p$	6.	$ap\lfloor q$	$\sim$	$a(p \parallel q)$
3.	$p+p$	$\sim$	p	7.	$(p+q)\lfloor r$	$\sim$	$p\lfloor r + q\lfloor r$
4.	$p+\mathbf{0}$	$\sim$	p	8.	$(p\lfloor q)\lfloor r$	$\sim$	$p\lfloor(q \parallel r)$
				9.	$p\lfloor\mathbf{0}$	$\sim$	p

For cases 1 through 7, denoting left and right sides by P and Q, one can show from the definition of $\xrightarrow{a}$ that $P \xrightarrow{a} P'$ iff $Q \xrightarrow{a} P'$; case 8 requires to be proven by induction on $|p|+|q|+|r|$ simultaneously with $(p \parallel q) \parallel r \sim p \parallel (q \parallel r)$, using the easily-proven $p \parallel q \sim q \parallel p$ (or see, e.g., [MIL80], **Theorem 5.5** *for a proof of the associativity of $\parallel$); case 9 requires a structural induction proof.* □

For the proof of completeness, we again define a *normal form* for expressions, and manipulate these to secure our result. However, its definition and some proofs in this section require a nontrivial complexity measure on the structure of terms, to justify that our inductively-defined concepts are well-defined, and that our inductive proofs are well-founded. The conditions which our complexity measure C must satisfy, as will become clear later, are as follows.

(i)	$C(at)$	$>$	$C(t)$;	(v)	$C(at\lfloor u)$	$>$	$C(t \parallel u)$;
(ii)	$C(t+u)$	$>$	$C(t)$;	(vi)	$C\big((t+u)\lfloor v\big)$	$>$	$C(t\lfloor v)$;
(iii)	$C(t+u)$	$>$	$C(u)$;	(vii)	$C\big((t+u)\lfloor v\big)$	$>$	$C(u\lfloor v)$;
(iv)	$C(x\lfloor t)$	$>$	$C(t)$;	$(viii)$	$C\big((t\lfloor u)\lfloor v\big)$	$>$	$C\big(t\lfloor(u \parallel v)\big)$.

With this in mind, let $width(\cdot)$ be defined as follows.

$$\begin{aligned} width(\mathbf{0}) &= 1; & width(at) &= 1 + width(t); \\ width(x) &= 1; & width(t+u) &= width(t) + width(u); \\ & & width(t \between u) &= width(t) + width(u). \end{aligned}$$

Thus informally $width(t)$ defines a certain size for the term t, namely, the number of $\{\mathbf{0}, x, a\}$ symbols appearing in t.

Next let the two norms $\|\cdot\|_0$ and $\|\cdot\|_1$ be defined as follows.

$$\begin{aligned} \|\mathbf{0}\|_0 &= 1; & \|at\|_0 &= 1 + \|t\|_0; \\ \|x\|_0 &= 1; & \|t+u\|_0 &= \max\big(\|t\|_0, \|u\|_0\big); \\ & & \|t \between u\|_0 &= \|t\|_0 + \|u\|_0 . \end{aligned}$$

$$\|t\|_1 = \begin{cases} width(t), & \text{if } t \text{ is not of the form } t_0 \between t_1, \\ width(t_0), & \text{if } t \text{ is of the form } t_0 \between t_1. \end{cases}$$

That is, $\|t\|_1$ gives the width of the term appearing to the left of the outermost $\between$ operator, if the outermost operator is $\between$; otherwise, $\|t\|_1$ gives the width of the whole term t. Then our complexity measure will be

$$\mathcal{C}(t) = \big(\|t\|_0, \|t\|_1\big),$$

a tuple ordered lexicographically. It is straightforward to check that this complexity measure $\mathcal{C}$ actually does satisfy the above inequalities.

Again we shall attack the problem of completeness by defining a denotation function for distinguishing between non-equivalent process terms, thus defining a unique normal form for any term. In this case though, the domain of values to which the denotation function maps terms of $\mathcal{P}_1$ is somewhat more complicated than before. However, it can be specified in the same way, as the least fixed point solution $\mathcal{D}_1$ to the set equation

$$\mathcal{D}_1 = \mathcal{P}_{FIN}\big(\mathbf{Var} \times \mathcal{D}_1 \ \cup \ \mathbf{Act} \times \mathcal{D}_1\big),$$

where again $\mathcal{P}_{FIN}(S)$ represents the set of finite subsets of S.

Definition 3.2.5 *The* denotation *of terms* $[\![\cdot]\!] : \mathcal{P}_1 \to \mathcal{D}_1$ *is defined inductively by case analysis of terms as follows.*

$$\begin{aligned}
[\![\mathbf{0}]\!] &= \emptyset & [\![\mathbf{0} \lfloor t]\!] &= \emptyset \\
[\![x]\!] &= \{(x, \emptyset)\} & [\![x \lfloor t]\!] &= \{(x, [\![t]\!])\} \\
[\![at]\!] &= \{(a, [\![t]\!])\} & [\![at \lfloor u]\!] &= \{(a, [\![t \parallel u]\!])\} \\
[\![t + u]\!] &= [\![t]\!] \cup [\![u]\!] & [\![(t + u) \lfloor v]\!] &= [\![t \lfloor v]\!] \cup [\![u \lfloor v]\!] \\
& & [\![(t \lfloor u) \lfloor v]\!] &= [\![t \lfloor (u \parallel v)]\!]
\end{aligned}$$

This is a valid inductive definition, due to our complexity measure defined above. Furthermore, we shall be justified in using the same case analysis in our inductive proofs to follow.

Informally, the denotation of a term is a set containing tuples which either represent the immediate actions which the term can perform along with the denotations of the resulting terms which the original term evolves into, or the unguarded variables appearing to the left of a $\lfloor$ operator together with the denotations of the terms appearing on the right side of the $\lfloor$.

We have a problem in our analysis to follow which derives from the confusion between the terms $(t \lfloor u) \lfloor v$ and $t \lfloor (u \parallel v)$. These two terms, although being defined to be semantically equivalent, give us trouble in our syntactic analysis. In particular,

$$(at \lfloor u) \lfloor v \xrightarrow{a} (t \parallel u) \parallel v,$$

whereas

$$at \lfloor (u \parallel v) \xrightarrow{a} t \parallel (u \parallel v) \neq (t \parallel u) \parallel v.$$

The problem arises as we have not enough power yet to express the fact that $[\![(t \parallel u) \parallel v]\!] = [\![t \parallel (u \parallel v)]\!]$. Also, we confuse in our analysis the two semantically equivalent terms x and $x \lfloor \mathbf{0}$. Problems arise in our analysis here when we have

$$x\{\bar{p}/\bar{x}\} \xrightarrow{a} p,$$

whereas

$$(x \lfloor\!\lfloor \mathbf{0})\{\bar{p}/\bar{x}\} \xrightarrow{a} p \parallel \mathbf{0} \neq p.$$

Again, we cannot yet prove that $[\![t \parallel \mathbf{0}]\!] = [\![t]\!]$.

To deal with these problems in our analysis in the remainder of this section, we shall continue to use $=$ to represent syntactic identity, and introduce $\equiv$ to represent syntactic identity modulo the following laws:

$$\begin{aligned}
&(C_1) \quad (x \parallel y) \parallel z \;=\; x \parallel (y \parallel z);\\
&(C_2) \quad x \parallel y \;=\; y \parallel x;\\
&(C_3) \quad x \parallel \mathbf{0} \;=\; x.
\end{aligned}$$

Notice that these are semantically sound laws, as they are all derivable from the sound set of laws $\mathcal{T}_1$, so $t \equiv u \implies t \sim u$.

The important properties which we shall use about the denotation of terms are the following.

Proposition 3.2.6

(i) $(a, T) \in [\![t]\!]$ *implies* $\exists t', t''$ *such that* $t \xrightarrow{a} t'$, $T = [\![t'']\!]$, *and* $t' \equiv t''$;

(ii) $t \xrightarrow{a} t'$ *implies* $\exists t'' \equiv t'$ *such that* $(a, [\![t'']\!]) \in [\![t]\!]$.

Proof:

By case analysis on the structure of t, *using induction on the structural complexity of* t. *We only demonstrate here the most difficult case of* $t = (u \lfloor\!\lfloor v) \lfloor\!\lfloor w$.

(i) $(a, T) \in [\![(u \lfloor\!\lfloor v) \lfloor\!\lfloor w]\!] \;=\; [\![u \lfloor\!\lfloor (v \parallel w)]\!]$

$\implies \exists t_0', t_0''$ *st* $u \lfloor\!\lfloor (v \parallel w) \xrightarrow{a} t_0'$, $T = \left[\!\left[t_0''\right]\!\right]$, *and* $t_0' \equiv t_0''$

(by the inductive hypothesis)

$$\Longrightarrow u \xrightarrow{a} u' \text{ st } t'_0 = u' \parallel (v \parallel w)$$

$$\Longrightarrow t \xrightarrow{a} (u' \parallel v) \parallel w \equiv t'_0 \equiv t''_0$$

$$\Longrightarrow \text{letting } t' = (u' \parallel v) \parallel w \text{ and } t'' = t''_0,$$

$$\text{we have } t \xrightarrow{a} t',\ T = [\![t'']\!],\ \text{and } t' \equiv t'';$$

(ii) $(u \lfloor\!\lfloor v) \lfloor\!\lfloor w \xrightarrow{a} t'$

$$\Longrightarrow u \xrightarrow{a} u' \text{ st } t' = (u' \parallel v) \parallel w$$

$$\Longrightarrow u \lfloor\!\lfloor (v \parallel w) \xrightarrow{a} u' \parallel (v \parallel w) \equiv t'$$

$$\Longrightarrow \exists t'' \equiv u' \parallel (v \parallel w) \equiv t' \text{ st } (a, [\![t'']\!]) \in [\![u \lfloor\!\lfloor (v \parallel w)]\!]$$

(by the inductive hypothesis)

But since $[\![(u \lfloor\!\lfloor v) \lfloor\!\lfloor w]\!] = [\![u \lfloor\!\lfloor (v \parallel w)]\!]$,

$$(a, [\![t'']\!]) \in [\![(u \lfloor\!\lfloor v) \lfloor\!\lfloor w)]\!].$$ □

Proposition 3.2.7

(i) $t \xrightarrow{a} t'$ *implies* $t\{\bar{p}/\bar{x}\} \xrightarrow{a} t'\{\bar{p}/\bar{x}\}$;

(ii) $(x, T) \in [\![t]\!]$ *and* $p_x \xrightarrow{a} p$ *implies* $\exists t', p'$ *such that* $t\{\bar{p}/\bar{x}\} \xrightarrow{a} p'$, $T = [\![t']\!]$, *and* $p' \equiv p \parallel t'\{\bar{p}/\bar{x}\}$.

Proof:

(i) *By structural induction on* t;

(ii) *By case analysis on the structure of* t, *using induction on the structural complexity of* t. *Again we only demonstrate here the most difficult case of* $t = (u \lfloor\!\lfloor v) \lfloor\!\lfloor w$;

$(x, T) \in [\![(u \lfloor\!\lfloor v) \lfloor\!\lfloor w]\!] = [\![u \lfloor\!\lfloor (v \parallel w)]\!]$ *and* $p_x \xrightarrow{a} p$

$$\Longrightarrow \exists t_0, p_0 \text{ st } u\{\bar{p}/\bar{x}\} \lfloor\!\lfloor \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \xrightarrow{a} p_0,$$

$$T = [\![t_0]\!],\ \text{and } p_0 \equiv p \parallel t_0\{\bar{p}/\bar{x}\};$$

(by the inductive hypothesis)

$$\Longrightarrow u\{\bar{p}/\bar{x}\} \xrightarrow{a} p'_0$$

$$\text{st } p_0 = p'_0 \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right);$$

$$\Longrightarrow t\{\bar{p}/\bar{x}\} \xrightarrow{a} \left(p'_0 \parallel v\{\bar{p}/\bar{x}\}\right) \parallel w\{\bar{p}/\bar{x}\} \equiv p_0;$$

Let $p' = \left(p'_0 \parallel v\{\bar{p}/\bar{x}\}\right) \parallel w\{\bar{p}/\bar{x}\}$ *and* $t' = t_0$;

Then $t\{\bar{p}/\bar{x}\} \xrightarrow{a} p'$, $T = [\![t']\!]$, *and* $p' \equiv p \parallel t'\{\bar{p}/\bar{x}\}$.

□

Proposition 3.2.8 $t\{\bar{p}/\bar{x}\} \xrightarrow{a} p$ *implies either*

(i) $\exists t'$ *such that* $t \xrightarrow{a} t'$ *and* $p \equiv t'\{\bar{p}/\bar{x}\}$; *or*

(ii) $\exists x, t', p'$ *such that* $(x, [\![t']\!]) \in [\![t]\!]$, $p_x \xrightarrow{a} p'$, *and* $p \equiv p' \parallel t'\{\bar{p}/\bar{x}\}$.

Proof:

By case analysis on the structure of t, *using induction on the structural complexity of* t. *Again we only demonstrate here the most difficult case of* $t = (u \mathbin{[\![} v) \mathbin{[\![} w$.

$$\left((u \mathbin{[\![} v) \mathbin{[\![} w\right)\{\bar{p}/\bar{x}\} = \left(u\{\bar{p}/\bar{x}\} \mathbin{[\![} v\{\bar{p}/\bar{x}\}\right) \mathbin{[\![} w\{\bar{p}/\bar{x}\} \xrightarrow{a} p$$

$$\Longrightarrow u\{\bar{p}/\bar{x}\} \xrightarrow{a} q \text{ st } p = \left(q \parallel v\{\bar{p}/\bar{x}\}\right) \parallel w\{\bar{p}/\bar{x}\};$$

$$\Longrightarrow u\{\bar{p}/\bar{x}\} \mathbin{[\![} \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right)$$

$$\xrightarrow{a} q \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \equiv p;$$

Thus by the inductive hypothesis, either

(i) $\exists t'_0$ *st* $u \mathbin{[\![} (v \parallel w) \xrightarrow{a} t'_0$

and $q \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \equiv t'_0\{\bar{p}/\bar{x}\}$

$$\Longrightarrow u \xrightarrow{a} u' \text{ st } t'_0 = u' \parallel (v \parallel w)$$

$$\Longrightarrow t \xrightarrow{a} (u' \parallel v) \parallel w \equiv t'_0$$

$$\Longrightarrow \exists t' \text{ st } t \xrightarrow{a} t' \text{ and } p \equiv t'\{\bar{p}/\bar{x}\};$$

or

(ii) $\exists x, t', p'$ *st* $(x, [\![t']\!]) \in [\![u \mathbin{[\![} (v \parallel w)]\!]$, $p_x \xrightarrow{a} p'$,

$$\text{and } q \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \equiv p' \parallel t'\{\bar{p}/\bar{x}\}$$

$$\Longrightarrow (x, [\![t']\!]) \in [\![t]\!],\ p_x \xrightarrow{a} p',$$

$$\text{and } p \equiv p' \parallel t'\{\bar{p}/\bar{x}\}. \qquad \square$$

Proposition 3.2.9 *Assume that* $|\mathbf{Act}| = \infty$*; let* $fv(t) \subseteq \bar{x} = \{x_1, x_2, ..., x_n\}$*, and* $\bar{p} = \{a_1.0, a_2.0, ..., a_n.0\}$*, where* $a_1, a_2, ..., a_n \in \mathbf{Act}$ *are distinct action symbols not appearing in* t*; then*

(i) $(x_k, T) \in [\![t]\!]$ *implies* $\exists p, t'$ *such that* $t\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p$, $T = [\![t']\!]$, *and* $p \equiv t'\{\bar{p}/\bar{x}\}$;

(ii) $t\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p$ *implies* $\exists t'$ *such that* $(x_k, [\![t']\!]) \in [\![t]\!]$ *and* $p \equiv t'\{\bar{p}/\bar{x}\}$.

Proof:

By case analysis on the structure of t*, using induction on the structural complexity of* t*. Again we only demonstrate the most difficult case of* $t = (u \lfloor\!\lfloor v) \lfloor\!\lfloor w)$.

(i) $(x_k, T) \in [\![(u \lfloor\!\lfloor v) \lfloor\!\lfloor w]\!] = [\![u \lfloor\!\lfloor (v \parallel w)]\!]$

$$\Longrightarrow \exists p, t' \text{ st } u\{\bar{p}/\bar{x}\} \lfloor\!\lfloor \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \xrightarrow{a_k} p,$$
$$T = [\![t']\!], \text{ and } p \equiv t'\{\bar{p}/\bar{x}\};$$

(by the inductive hypothesis)

$$\Longrightarrow u\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p' \text{ st } p = p' \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right)$$
$$\Longrightarrow t\{\bar{p}/\bar{x}\} \xrightarrow{a_k} \left(p' \parallel v\{\bar{p}/\bar{x}\}\right) \parallel w\{\bar{p}/\bar{x}\} \equiv p;$$

(ii) $\left((u \lfloor\!\lfloor v) \lfloor\!\lfloor w\right)\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p$

$$\Longrightarrow u\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p' \text{ st } p = \left(p' \parallel v\{\bar{p}/\bar{x}\}\right) \parallel w\{\bar{p}/\bar{x}\}$$
$$\Longrightarrow u\{\bar{p}/\bar{x}\} \lfloor\!\lfloor \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right)$$
$$\xrightarrow{a_k} p' \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \equiv p$$
$$\Longrightarrow \exists t' \text{ st } (x_k, [\![t']\!]) \in [\![u \lfloor\!\lfloor (v \parallel w)]\!]$$
$$\text{and } p' \parallel \left(v\{\bar{p}/\bar{x}\} \parallel w\{\bar{p}/\bar{x}\}\right) \equiv t'\{\bar{p}/\bar{x}\}$$

(by the inductive hypothesis)

$$\Longrightarrow (x_k, [\![t']\!]) \in [\![t]\!] \text{ and } p \equiv t'\left\{\bar{p}/\bar{x}\right\}. \qquad \square$$

The denotations of terms are again used to define the *normal form* of a term, which in this case will be an equivalent term which is expressed as a sum of action-prefixed normal form terms added to a sum of left merge terms whose left operands are variables. The normal form of a term is extracted from its denotation as follows.

Definition 3.2.10 *The* normal form *of terms $nf(\cdot)$ is given by:*

$$nf(t) \;=\; \sigma([\![t]\!]),$$

where

$$\sigma(T) \;=\; \sum_{(a,S)\in T} a.\sigma(S) \;+\; \sum_{(x,S)\in T} x\lfloor\sigma(S).$$

Once again, by convention we let $\sigma(\emptyset) = \mathbf{0}$.

Proposition 3.2.11 $\mathcal{T}_1 \vdash t = nf(t)$.

Proof:

By induction on the complexity of the structure of t, using the axioms of $\mathcal{T}_1$ and the definitions of $[\![\cdot]\!]$ and $\sigma(\cdot)$.

- $\mathbf{0} = \sigma(\emptyset) = \sigma([\![\mathbf{0}]\!]) = nf(\mathbf{0})$;
- $x =_{\mathcal{T}_1} x\lfloor\mathbf{0} = \sigma(\{(x,\emptyset)\}) = \sigma([\![x]\!]) = nf(x)$;
- $\mathcal{T}_1 \vdash t = nf(t) = \sigma([\![t]\!])$
  $\Longrightarrow at =_{\mathcal{T}_1} a(\sigma([\![t]\!]))$
  $= \sigma(\{(a, [\![t]\!])\})$
  $= \sigma([\![at]\!]) = nf(at)$;
- $\mathcal{T}_1 \vdash t = nf(t) = \sigma([\![t]\!])$, $\mathcal{T}_1 \vdash u = nf(u) = \sigma([\![u]\!])$

$$\begin{aligned}\Longrightarrow\ t+u &=_{\mathcal{T}_1} \sigma([\![t]\!])+\sigma([\![u]\!])\\ &=_{\mathcal{T}_1} \sigma([\![t]\!]\cup[\![u]\!])\\ &=\sigma([\![t+u]\!])=nf(t+u);\end{aligned}$$

- $\mathbf{0}\lfloor t =_{\mathcal{T}_1} \mathbf{0}=\sigma(\emptyset)=\sigma([\![\mathbf{0}\lfloor t]\!])=nf(\mathbf{0}\lfloor t)$;

- $\mathcal{T}_1\vdash t\lfloor u=nf(t\lfloor u)=\sigma([\![t\lfloor u]\!])$,
  $\mathcal{T}_1\vdash u\lfloor t=nf(u\lfloor t)=\sigma([\![u\lfloor t]\!])$
  $$\begin{aligned}\Longrightarrow\ at\lfloor u &=_{\mathcal{T}_1} a(t\lfloor u+u\lfloor t)\\ &=_{\mathcal{T}_1} a\Big(\sigma([\![t\lfloor u]\!])+\sigma([\![u\lfloor t]\!])\Big)\\ &=_{\mathcal{T}_1} a\Big(\sigma([\![t\lfloor u]\!]\cup[\![u\lfloor t]\!])\Big)\\ &=a\Big(\sigma([\![t\lfloor u+u\lfloor t]\!])\Big)\\ &=a\Big(\sigma([\![t\parallel u]\!])\Big)\\ &=\sigma(\{(a,[\![t\parallel u]\!])\})\\ &=\sigma([\![at\lfloor u]\!])=nf(at\lfloor u);\end{aligned}$$

- $\mathcal{T}_1\vdash t=nf(t)=\sigma([\![t]\!])$
  $$\begin{aligned}\Longrightarrow\ x\lfloor t &=_{\mathcal{T}_1} x\lfloor\sigma([\![t]\!])\\ &=\sigma(\{x,[\![t]\!]\})\\ &=\sigma([\![x\lfloor t]\!])=nf(x\lfloor t);\end{aligned}$$

- $\mathcal{T}_1\vdash t\lfloor v=nf(t\lfloor v)=\sigma([\![t\lfloor v]\!])$,
  $\mathcal{T}_1\vdash u\lfloor v=nf(u\lfloor v)=\sigma([\![u\lfloor v]\!])$
  $$\begin{aligned}\Longrightarrow\ (t+u)\lfloor v &=_{\mathcal{T}_1} t\lfloor v+u\lfloor v\\ &=_{\mathcal{T}_1} \sigma([\![t\lfloor v]\!])+\sigma([\![u\lfloor v]\!])\\ &=_{\mathcal{T}_1} \sigma([\![t\lfloor v]\!]\cup[\![u\lfloor v]\!])\\ &=\sigma([\![(t+u)\lfloor v]\!])=nf((t+u)\lfloor v);\end{aligned}$$

- $\mathcal{T}_1\vdash t\lfloor(u\parallel v)=nf(t\lfloor(u\parallel v))=\sigma([\![t\lfloor(u\parallel v)]\!])$
  $$\begin{aligned}\Longrightarrow\ (t\lfloor u)\lfloor v &=_{\mathcal{T}_1} t\lfloor(u\parallel v)\\ &=_{\mathcal{T}_1} \sigma([\![t\lfloor(u\parallel v)]\!])\\ &=\sigma([\![(t\lfloor u)\lfloor v]\!])=nf((t\lfloor u)\lfloor v).\end{aligned}$$ □

Corollary 3.2.12 $[\![t]\!]=[\![u]\!]\Longrightarrow\mathcal{T}_1\vdash t=u$.

Proof:

$$t =_{\mathcal{T}_1} nf(t) = \sigma([\![t]\!]) = \sigma([\![u]\!]) = nf(u) =_{\mathcal{T}_1} u. \qquad \square$$

For the proof of the following proposition, we need to invoke an inductive argument on the *sizes* of the denotation sets, where the size is basically the *rank* of the set. For this we make the following definition.

Definition 3.2.13 *For* $S \in \mathcal{D}_1$,

$$rank(S) = \begin{cases} 0, & \text{if } S = \emptyset, \\ 1 + \max\{rank(T) \mid (a,T) \in S \vee (x,T) \in S\}, & \text{otherwise.} \end{cases}$$

Proposition 3.2.14 *Suppose that* $[\![t]\!] \neq [\![u]\!]$; *assume further that* $|\mathbf{Act}| = \infty$; *let* $fv(t) \cup fv(u) \subseteq \bar{x} = \{x_1, x_2, ..., x_n\}$, *and* $\bar{p} = \{a_1.\mathbf{0}, a_2.\mathbf{0}, ..., a_n.\mathbf{0}\}$, *where* $a_1, a_2, ..., a_n \in \mathbf{Act}$ *are distinct action symbols not appearing in* u *or* v; *then* $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$.

Proof:

By induction on rank($[\![t]\!]$ *). Suppose* $[\![t]\!] \neq [\![u]\!]$, *and let* $\bar{x}, \bar{p}$ *be as given in the proposition; there are four cases to consider:* $(x,T) \in [\![t]\!] \setminus [\![u]\!]$, $(x,T) \in [\![u]\!] \setminus [\![t]\!]$, $(a,T) \in [\![t]\!] \setminus [\![u]\!]$, $(a,T) \in [\![u]\!] \setminus [\![t]\!]$:

- *Suppose* $(x_k,T) \in [\![t]\!]$, *but* $(x_k,T) \notin [\![u]\!]$;

  then by **Proposition 3.2.9(i)**, $\exists p, t'$ *st* $t\{\bar{p}/\bar{x}\} \xrightarrow{a_k} p$,
  $T = [\![t']\!]$, *and* $p \equiv t'\{\bar{p}/\bar{x}\}$;
  Suppose $t\{\bar{p}/\bar{x}\} \sim u\{\bar{p}/\bar{x}\}$;
  Then $\exists q \sim t'\{\bar{p}/\bar{x}\}$ *st* $u\{\bar{p}/\bar{x}\} \xrightarrow{a_k} q$;
  so by **Proposition 3.2.9(ii)**, $\exists u'$ *st* $(x_k, [\![u']\!]) \in [\![u]\!]$
  and $q \equiv u'\{\bar{p}/\bar{x}\} \sim t'\{\bar{p}/\bar{x}\}$;
  but $(x_k, [\![t']\!]) \notin [\![u]\!] \implies [\![t']\!] \neq [\![u']\!]$;
  so by the induction hypothesis, $t'\{\bar{p}/\bar{x}\} \not\sim u'\{\bar{p}/\bar{x}\}$
  (contradiction);
  thus $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Similarly,* $(x_k, T) \in [\![u]\!]$ *and* $(x_k, T) \notin [\![t]\!] \implies t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Suppose* $(a, T) \in [\![t]\!]$, *but* $(a, T) \notin [\![u]\!]$;

  then by **Proposition 3.2.6(i)**, $\exists t', t''$ *st* $t \xrightarrow{a} t'$,
  $T = [\![t'']\!]$, *and* $t' \equiv t''$;
  thus by **Proposition 3.2.7(i)**, $t\{\bar{p}/\bar{x}\} \xrightarrow{a} t'\{\bar{p}/\bar{x}\}$;
  Suppose $t\{\bar{p}/\bar{x}\} \sim u\{\bar{p}/\bar{x}\}$;
  Then $\exists p \sim t'\{\bar{p}/\bar{x}\}$ *st* $u\{\bar{p}/\bar{x}\} \xrightarrow{a} p$;
  hence by **Proposition 3.2.8**, *either:*
  (i) $\exists u'$ *st* $u \xrightarrow{a} u'$ *and* $p \equiv u'\{\bar{p}/\bar{x}\} \sim t'\{\bar{p}/\bar{x}\}$;
  hence by **Proposition 3.2.6(ii)**,
  $\exists u'' \equiv u'$ *st* $(a, [\![u'']\!]) \in [\![u]\!]$;
  thus since $(a, [\![t']\!]) \notin [\![u]\!]$, $[\![t']\!] \neq [\![u'']\!]$;
  so by the inductive hypothesis,
  $t'\{\bar{p}/\bar{x}\} \not\sim u''\{\bar{p}/\bar{x}\} \sim u'\{\bar{p}/\bar{x}\}$;
  (contradiction); *or*
  (ii) $\exists x, u', p'$ *st* $(x, [\![u']\!]) \in [\![u]\!]$, $p_x \xrightarrow{a} p'$,
  and $p \equiv p' \parallel u'\{\bar{p}/\bar{x}\}$;
  i.e., $\exists j, u'$ *st* $(x_j, [\![u']\!]) \in [\![u]\!]$, $a = a_j$,
  and $p \equiv u'\{\bar{p}/\bar{x}\} \sim t'\{\bar{p}/\bar{x}\}$;
  but by assumption, a_j *does not appear in* t;
  hence $(a_j, T) \notin [\![t]\!]$ *for any* T *(contradiction);*
  thus $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$;

- *Similarly,* $(a, T) \in [\![u]\!]$ *and* $(a, T) \notin [\![t]\!] \implies t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$.

□

Corollary 3.2.15 $t \sim u \implies [\![t]\!] = [\![u]\!]$.

Proof:

Suppose $[\![t]\!] \neq [\![u]\!]$; *let* $\bar{x}$, $\bar{p}$ *be as in* **Proposition 3.2.14**;

then $t\{\bar{p}/\bar{x}\} \not\sim u\{\bar{p}/\bar{x}\}$*; hence by the ω-completeness definition of congruence of open terms,* $t \not\sim u$. □

Corollary 3.2.16 (ω-Completeness) $t \sim u \implies \mathcal{T}_1 \vdash t = u$.

Proof:

Follows from **Corollary 3.2.15** *and* **Corollary 3.2.12.** □

As one last note about this sublanguage $\mathcal{P}_1$, we can come up with an alternative characterisation for the extensional observational congruence similar to that presented in **Proposition 3.1.15**, by incorporating the effect of unguarded variables in terms into the definition of the bisimulation underlying the congruence. However in this case, we must be more careful in our approach. With the sublanguage $\mathcal{P}_0$ of nondeterministic terms, once a variable process was started, all other subterms were ignored. However with the terms in this sublanguage $\mathcal{P}_1$, we must account for the subterms representing processes which run concurrently with variable processes which may be started. Thus we redefine the notion of unguarded variables in this case to account for the extra required information as follows:

Definition 3.2.17 *The* extended unguarded variable occurrences $\widehat{\mathcal{U}}(\cdot)$ *of terms are defined inductively on the complexity of the structure of terms as follows:*

$$\begin{aligned}
\widehat{\mathcal{U}}(\mathbf{0}) &= \emptyset; & \widehat{\mathcal{U}}(\mathbf{0}\llbracket v) &= \emptyset;\\
\widehat{\mathcal{U}}(x) &= \{(x,\mathbf{0})\}; & \widehat{\mathcal{U}}(x\llbracket v) &= \{(x,v)\};\\
\widehat{\mathcal{U}}(at) &= \emptyset; & \widehat{\mathcal{U}}(at\llbracket v) &= \emptyset;\\
\widehat{\mathcal{U}}(t+u) &= \widehat{\mathcal{U}}(t)\cup\widehat{\mathcal{U}}(u); & \widehat{\mathcal{U}}\big((t+u)\llbracket v\big) &= \widehat{\mathcal{U}}(t\llbracket v)\cup\widehat{\mathcal{U}}(u\llbracket v);\\
& & \widehat{\mathcal{U}}\big((t\llbracket u)\llbracket v\big) &= \widehat{\mathcal{U}}\big(t\llbracket(u\parallel v)\big).
\end{aligned}$$

Our alternate characterisation of our semantic congruence of open terms will then be given by the following definition.

Definition 3.2.18 $t \cong u$ *iff for all* $a \in \mathbf{Act}$*:*

(i) $t \xrightarrow{a} t' \implies \exists u' \text{ st } u \xrightarrow{a} u' \text{ and } t' \cong u';$

(ii) $u \xrightarrow{a} u' \implies \exists t' \text{ st } t \xrightarrow{a} t' \text{ and } t' \cong u';$

(iii) $(x,t') \in \widehat{\mathcal{U}}(t) \implies \exists u' \text{ st } (x,u') \in \widehat{\mathcal{U}}(u) \text{ and } t' \cong u';$

(iv) $(x,u') \in \widehat{\mathcal{U}}(u) \implies \exists t' \text{ st } (x,t') \in \widehat{\mathcal{U}}(t) \text{ and } t' \cong u'.$

That this definition gives us our required congruence comes as a corollary of the following technical propositions.

Proposition 3.2.19 $[\![t]\!] = \left\{(a,[\![t']\!]) \;\middle|\; t \xrightarrow{a} t'\right\} \cup \left\{(x,[\![t']\!]) \;\middle|\; (x,t') \in \widehat{\mathcal{U}}(t)\right\}.$

Proof:

By induction on the structural complexity of t*, using the definitions of* $\xrightarrow{a}$ *and* $\widehat{\mathcal{U}}$*.* □

Proposition 3.2.20 $[\![t]\!] = [\![u]\!]$ *iff* $t \cong u$.

Proof:

Using **Proposition 3.2.19**, *by induction on the structural complexity of* t*.* □

Corollary 3.2.21 $t \cong u$ *iff* $t \sim u$.

Proof:

($\Longrightarrow$) *Follows from* **Proposition 3.2.20**, **Corollary 3.2.12** *and the Soundness* **Proposition 3.2.4**.

($\Longleftarrow$) *Follows from* **Corollary 3.2.15** *and* **Proposition 3.2.20**. □

3.3 Concurrency With The Full Merge Operator

The next language $\mathcal{P}_2$ we consider is a language which does not contain the left merge operator $\lfloor\!\lfloor$ in its signature, but rather only the full merge operator $\|$, which was a derived operator in the language $\mathcal{P}_1$. The language is defined by the signature $\Sigma_2 = \{\mathbf{0}, ., +, \| \}$. This is the most basic language of nondeterminism and concurrency used in CCS, but as we shall see, the difficult problems which this thesis is addressing are confronted in this most simple language. Namely, we shall demonstrate the difficulty in finding an ω-complete set of equational axioms for this language under strong observational congruence (and will in fact leave this problem open), and will in a later chapter actually show that any complete (not simply just ω-complete) axiomatisation must be infinite.

Again the semantic equivalence which we consider here is the strong observational congruence $\sim$ defined above. This theory is completely characterised by the following (infinite) set of axioms.

$$(A_1)\quad (x+y)+z = x+(y+z); \qquad (A_3)\quad x+x = x;$$
$$(A_2)\quad x+y = y+x; \qquad (A_4)\quad x+\mathbf{0} = x;$$

$$(Exp_{mn})\ For\ u = \sum_{i=1}^{m}\alpha_i x_i\ and\ v = \sum_{j=1}^{n}\beta_j y_j,$$

$$u \,\|\, v = \sum_{i=1}^{m}\alpha_i(x_i \,\|\, v) + \sum_{j=1}^{n}\beta_j(u \,\|\, y_j).$$

Proposition 3.3.1 *The strong observational congruence over closed terms of the language $\mathcal{P}_2$ is exactly the congruence induced by the infinite axiom set*

$$\mathcal{T}_2^0 = \Big\{(A_1),(A_2),(A_3),(A_4)\Big\} \cup \Big\{(Exp_{mn}) \;\Big|\; m,n \geq 0\Big\}.$$

Proof:

For a full proof, see, e.g., [HEN85], **Theorem 4.1**. *Otherwise note that to prove* $p \sim q$, *we need simply eliminate all occurrences of the full merge operator* $\|$ *from each of* p *and* q *using the (valid) Expansion Theorem axioms* (Exp_{mn}), *and prove the remaining terms (from the sublanguage* $\mathcal{P}_0^0$*) to be equal using the sum laws* (A_1)-(A_4), *which we proved in* **Section 3.1** *to be complete for reasoning about* $\mathcal{P}_0$ *terms.*

□

However, these axioms do not suffice as an ω-complete set of equational axioms for our congruence. For example, as pointed out in [HEN85], these axioms cannot prove the following valid laws.

$$(C_1) \quad (x \,\|\, y) \,\|\, z \;=\; x \,\|\, (y \,\|\, z);$$
$$(C_2) \quad x \,\|\, y \;=\; y \,\|\, x;$$
$$(C_3) \quad x \,\|\, \mathbf{0} \;=\; x.$$

In order to prove these using the theory $\mathcal{T}_2^0$, we would need to invoke a structural induction argument. However, we could easily produce a (counter-) model for the equational theory $\mathcal{T}_2^0$ which would contradict each of these statements.

Another simple law which cannot be proven within $\mathcal{T}_2^0$, even with the above three laws (C_1), (C_2) and (C_3) is the following Absorption Law expressing the partial application of the Expansion Theorem axioms (Exp_{mn}).

$$(Abs) \quad (\alpha x + y) \,\|\, z \;=\; (\alpha x + y) \,\|\, z \;+\; \alpha(x \,\|\, z).$$

That this law is sound can again be proven by structural induction using the theory $\mathcal{T}_2^0$; it is a straightforward corollary of the Expansion Theorem laws (Exp_{mn}) and the +-idempotence law (A_3). However again we could produce a countermodel satisfying the theory $\mathcal{T}_2^0 \cup \{(C_1), (C_2), (C_3)\}$ but contradicting (Abs), demonstrating that $\mathcal{T}_2^0 \cup \{(C_1), (C_2), (C_3)\} \not\vdash (Abs)$.

In actual fact, there are seemingly arbitrarily-complex equational axioms which are independent and which therefore must be included in any ω-complete set of axioms. For instance consider the following *reduction* laws which can all be seen to be valid ∼-equivalence laws, but which cannot be proven within $\mathcal{T}_2^0$.

$$\alpha x \parallel (y+z) \;+\; \alpha(x \parallel y) \;+\; \alpha(x \parallel z)$$
$$= \; \alpha\big(x \parallel (y+z)\big) \;+\; \alpha x \parallel y \;+\; \alpha x \parallel z;$$

$$(x+v) \parallel (y+z) \;+\; x \parallel y \;+\; x \parallel z \;+\; v \parallel y \;+\; v \parallel z$$
$$= \; x \parallel (y+z) \;+\; v \parallel (y+z) \;+\; (x+v) \parallel y \;+\; (x+v) \parallel z;$$

$$(\alpha x+v) \parallel (\beta y+z) \;+\; \alpha(x \parallel z) \;+\; \beta(v \parallel y) \;+\; v \parallel z$$
$$= \; \alpha\big(x \parallel (\beta y+z)\big) \;+\; \beta\big((\alpha x+v) \parallel y\big)$$
$$+\, v \parallel (\beta y+z) \;+\; (\alpha x+v) \parallel z;$$

$$(\alpha x+v) \parallel (y+z+u) \;+\; \alpha(x \parallel y) \;+\; \alpha(x \parallel z) \;+\; \alpha(x \parallel u)$$
$$+\, v \parallel y \;+\; v \parallel z \;+\; v \parallel u$$
$$= \; \alpha\big(x \parallel (y+z+u)\big) \;+\; v \parallel (y+z+u)$$
$$+\, (\alpha x+v) \parallel y \;+\; (\alpha x+v) \parallel z \;+\; (\alpha x+v) \parallel u;$$

etc.

Each law in this series demonstrates how to express one term which contains a single *"largest"* parallel composition as a summand (the first summand on the left hand side of each equation) as a sum of terms containing only smaller parallel combinations. In each case, in some sense only the minimum amount of *"excess baggage"* is included in the law in order to reduce the large composition. For instance, the third law above demonstrates the least (information-theoretic) valid statement expressing the parallel composition $(\alpha x + v) \parallel (\beta y + z)$ added to terms involving only smaller parallel compositions as a sum of strictly smaller compositions. That is to say, any valid statement which does express the parallel composition $(\alpha x + v) \parallel (\beta y + z)$ added to terms involving only smaller parallel compositions as a sum of strictly smaller compositions must contain on each side of the equation at least as much single-step behaviour as specified by the law above; the left and right hand sides of the equation must have at least the abilities to proceed as those of the minimal law above.

Each different pair of sumforms taken as operands to the parallel operator produces a new law independent of those given by less complicated sumform terms placed in parallel. For instance, the third law in the above list is not simply an instance of the second law, by replacing x by αx and y by βy. This is so as more information about an indeterminate process gives rise to a more specific law.

Notice that when one of the factors in the term wanting to be reduced out is a simple variable, then the reduction cannot be successful. We cannot express such a parallel composition, added to only simpler parallel compositions, as a sum of strictly smaller compositions. The large composition will in fact have to appear on both sides of the equality sign. The simplest such case of this phenomena was already encountered with the Absorption Law (*Abs*) above.

The degenerate case, when all of the summands in the two terms to be combined in parallel are action-prefixed terms, reduces to the Expansion Theorem: no terms have to be added to the composition, which is equated to its expanded version. In fact, the complete sequence of laws can be generalised into an axiom schema similar to that for the Expansion Theorem in the following way.

$$\begin{aligned} \textit{For } P &= \sum_{i=1}^{m} \alpha_i u_i + \sum_{j=1}^{r} x_j \\ \textit{and } Q &= \sum_{i=1}^{n} \beta_i v_i + \sum_{j=1}^{s} y_j, \end{aligned}$$

$$\begin{aligned} P \parallel Q + \sum_{i=1}^{m}\sum_{j=1}^{s} \alpha_i(u_i \parallel y_j) + \sum_{i=1}^{n}\sum_{j=1}^{r} \beta_i(v_i \parallel x_j) + \sum_{i=1}^{r}\sum_{j=1}^{s} x_i \parallel y_j \\ = \sum_{i=1}^{m} \alpha_i(u_i \parallel Q) + \sum_{i=1}^{n} \beta_i(P \parallel v_i) \\ + \sum_{j=1}^{r} x_j \parallel Q + \sum_{j=1}^{s} P \parallel y_j. \end{aligned}$$

Notice that this schema includes the Expansion Theorem axioms (Exp_{mn}), the degenerate case of this sequence (by setting $r = s = 0$ in the definition of P and Q), but it does *not* include the Absorption Law (*Abs*). This is because the law (*Abs*) is an *absorption* law and not a *reduction* law; both sides of the law contain the largest parallel composition. As remarked above, since one of the factors in the

largest composition in the law (*Abs*) is a simple variable, both sides of the equality must contain the composition. Since this sequence of laws only attempts to present the least amount of smaller compositions to use to express a term containing the large composition as a term not containing the composition, nothing is implied by this sequence when this is not possible. In fact, the closest we can get to (*Abs*) using this sequence is the following reflexive identity (by setting $m = n = r = 1$ and $s = 0$ in the definitions of P and Q in the axioms schema).

$$(\alpha x + y) \parallel z \;+\; \alpha(x \parallel z) \;+\; y \parallel z$$
$$= \;(\alpha x + y) \parallel z \;+\; \alpha(x \parallel z) \;+\; y \parallel z.$$

The above axiom schema only deals with the case when two terms are combined in parallel. There are analogous axioms which are independent of the above dealing with three processes combined in parallel. For example, we have the following three-factor reduction laws.

$$\begin{aligned}
&\alpha x \parallel (y + z) \parallel (u + v) \;+\; \alpha\big(x \parallel y \parallel (u + v)\big) \;+\; \alpha\big(x \parallel z \parallel (u + v)\big) \\
&\qquad + \alpha\big(x \parallel (y + z) \parallel u\big) \;+\; \alpha\big(x \parallel (y + z) \parallel v\big) \\
&\qquad + \alpha x \parallel y \parallel u \;+\; \alpha x \parallel y \parallel v \;+\; \alpha x \parallel z \parallel u \;+\; \alpha x \parallel z \parallel v \\
&= \alpha\big(x \parallel (y + z) \parallel (u + v)\big) \;+\; \alpha(x \parallel y \parallel u) \;+\; \alpha(x \parallel y \parallel v) \\
&\qquad + \alpha(x \parallel z \parallel u) \;+\; \alpha(x \parallel z \parallel v) \\
&\qquad + \alpha x \parallel y \parallel (u + v) \;+\; \alpha x \parallel z \parallel (u + v) \\
&\qquad + \alpha x \parallel (y + z) \parallel u \;+\; \alpha x \parallel (y + z) \parallel v;
\end{aligned}$$

$$\begin{aligned}
&\alpha x \parallel (\beta y + z) \parallel (u + v) \;+\; \alpha\big(x \parallel z \parallel (u + v)\big) \;+\; \alpha\big(x \parallel (\beta y + z) \parallel u\big) \\
&\qquad + \alpha\big(x \parallel (\beta y + z) \parallel v\big) \;+\; \alpha x \parallel z \parallel u \;+\; \alpha x \parallel z \parallel v \\
&\qquad + \beta(\alpha x \parallel y \parallel u) \;+\; \beta(\alpha x \parallel y \parallel v) \\
&= \alpha\big(x \parallel (\beta y + z) \parallel (u + v)\big) \;+\; \beta\big(\alpha x \parallel y \parallel (u + v)\big) \\
&\qquad + \alpha(x \parallel z \parallel u) \;+\; \alpha(x \parallel z \parallel v)
\end{aligned}$$

$$+ \alpha x \parallel z \parallel (u + v) \; + \; \alpha x \parallel (\beta y + z) \parallel u$$

$$+ \alpha x \parallel (\beta y + z) \parallel v;$$

etc.

Again these axioms give the minimum laws for reducing large parallel compositions, and as such are not direct consequences of the two-factor laws. These axioms for three terms combined in parallel can be generalised into an axiom schema similarly to the above in the following way.

$$\begin{aligned} \textit{For } P &= \sum_{i=1}^{m_p} \alpha_i u_i \; + \; \sum_{j=1}^{n_p} x_j \\ \textit{and } Q &= \sum_{i=1}^{m_q} \beta_i v_i \; + \; \sum_{j=1}^{n_q} y_j \\ \textit{and } R &= \sum_{i=1}^{m_r} \sigma_i w_i \; + \; \sum_{j=1}^{n_r} z_j, \end{aligned}$$

$$\begin{aligned} P \parallel Q \parallel R \; &+ \; \sum_{i=1}^{m_p}\sum_{j=1}^{n_q} \alpha_i(u_i \parallel y_j \parallel R) \; + \; \sum_{i=1}^{m_p}\sum_{j=1}^{n_r} \alpha_i(u_i \parallel Q \parallel z_j) \\ &+ \; \sum_{i=1}^{m_q}\sum_{j=1}^{n_p} \beta_i(x_j \parallel v_i \parallel R) \; + \; \sum_{i=1}^{m_q}\sum_{j=1}^{n_r} \beta_i(P \parallel v_i \parallel z_j) \\ &+ \; \sum_{i=1}^{m_r}\sum_{j=1}^{n_p} \sigma_i(x_j \parallel Q \parallel w_i) \; + \; \sum_{i=1}^{m_r}\sum_{j=1}^{n_q} \sigma_i(P \parallel y_j \parallel w_i) \\ &+ \; \sum_{i=1}^{n_q}\sum_{j=1}^{n_r} P \parallel y_i \parallel z_j \; + \; \sum_{i=1}^{n_p}\sum_{j=1}^{n_r} x_i \parallel Q \parallel z_j \; + \; \sum_{i=1}^{n_p}\sum_{j=1}^{n_q} x_i \parallel y_j \parallel R \\ = \; &\sum_{i=1}^{m_p} \alpha_i(u_i \parallel Q \parallel R) \; + \; \sum_{i=1}^{m_q} \beta_i(P \parallel v_i \parallel R) \; + \; \sum_{i=1}^{m_r} \sigma_i(P \parallel Q \parallel w_i) \\ &+ \; \sum_{j=1}^{n_p} x_j \parallel Q \parallel R \; + \; \sum_{j=1}^{n_q} P \parallel y_j \parallel R \; + \; \sum_{j=1}^{n_r} P \parallel Q \parallel z_j \\ &+ \; \sum_{i=1}^{m_p}\sum_{j=1}^{n_q}\sum_{k=1}^{n_r} \alpha_i(u_i \parallel y_j \parallel z_k) \; + \; \sum_{i=1}^{m_q}\sum_{j=1}^{n_p}\sum_{k=1}^{n_r} \beta_i(x_j \parallel v_i \parallel z_k) \\ &+ \; \sum_{i=1}^{m_r}\sum_{j=1}^{n_p}\sum_{k=1}^{n_q} \sigma_i(x_j \parallel y_k \parallel w_i) \; + \; \sum_{i=1}^{n_p}\sum_{j=1}^{n_q}\sum_{k=1}^{n_r} x_i \parallel y_j \parallel z_k. \end{aligned}$$

Similarly, there are axioms like these for n processes combined in parallel, for each $n > 1$, none of which are instances of the others, and characterised by greatly

increasingly complex axiom schemata as n increases. However, all of these laws can be collected together into one terse (albeit totally unreadable) axiom schema as follows:

$(Red_{t,m_1,n_1,m_2,n_2,\ldots,m_t,n_t})$.

$$For\ P_i = \sum_{j=1}^{m_i} \alpha_{ij} u_{ij} + \sum_{j=1}^{n_i} x_{ij} \quad (\text{for } i = 1,2,\ldots,t),$$

$$\sum_{\substack{I\subseteq\{1,2,\ldots,t\} \\ |I| \text{ even}}} \sum_{\substack{\sigma:I\to\omega \\ 1\le\sigma_i\le n_i}} \Big(\prod_{i\in I} x_{i\sigma_i} \parallel \prod_{i\notin I} P_i \Big)$$

$$+ \sum_{\substack{I\subseteq\{1,2,\ldots,t\} \\ |I| \text{ even}}} \sum_{i\in I} \sum_{1\le j\le m_i} \sum_{\substack{\sigma:I\setminus\{i\}\to\omega \\ 1\le\sigma_k\le n_k}} \alpha_{ij} \Big(u_{ij} \parallel \prod_{k\in I\setminus\{i\}} x_{k\sigma_k} \parallel \prod_{s\notin I} P_s \Big)$$

$$= \sum_{\substack{I\subseteq\{1,2,\ldots,t\} \\ |I| \text{ odd}}} \sum_{\substack{\sigma:I\to\omega \\ 1\le\sigma_i\le n_i}} \Big(\prod_{i\in I} x_{i\sigma_i} \parallel \prod_{i\notin I} P_i \Big)$$

$$+ \sum_{\substack{I\subseteq\{1,2,\ldots,t\} \\ |I| \text{ odd}}} \sum_{i\in I} \sum_{1\le j\le m_i} \sum_{\substack{\sigma:I\setminus\{i\}\to\omega \\ 1\le\sigma_k\le n_k}} \alpha_{ij} \Big(u_{ij} \parallel \prod_{k\in I\setminus\{i\}} x_{k\sigma_k} \parallel \prod_{s\notin I} P_s \Big).$$

Notice here that $Exp_{mn} = (Red_{2,m,0,n,0})$.

Thus we have an abundance of valid equations which we cannot prove within our theory $\mathcal{T}_2^0$ and which are all seemingly mutually independent. Hence these would all need to be added to our theory $\mathcal{T}_2^0$ to approximate an ω-complete theory $\mathcal{T}_2$ for strong observational congruence over $\mathcal{P}_2$. However, this is still not enough.

In all of the above reduction laws, we show how to reduce a term containing a large parallel composition into one not containing the composition, by using only strictly smaller compositions. There is also an abundance of laws which show under what conditions one (open) term can be absorbed into another (possibly more complicated) term. That is, when terms P and Q are such that $P + Q = Q$, whence we say that P is *absorbed into* Q. These arise when the capabilities of some term to proceed are matched completely by a subset of the capabilities of another term; in this case, the former term can be absorbed into the latter. For instance we have the following law.

$$x_1 \parallel (y_1 + y_2) \;+\; (x_1 + x_2) \parallel y_1 \;+\; x_1 \parallel y_1$$
$$= \; x_1 \parallel (y_1 + y_2) \;+\; (x_1 + x_2) \parallel y_1.$$

Here, the $x_1 \parallel y_1$ term is absorbed by the two terms to which it is added. The motivation for suspecting this to be a valid law comes from the following reasoning: the possibility of the absorbed term $x_1 \parallel y_1$ on the left hand side of the equation proceeding via the process represented by the subterm x_1 is matched on the right hand side of the equation by the possibility of proceeding with the $(x_1 + x_2) \parallel y_1$ term via the same indeterminate process x_1. Similarly, the possibility of the absorbed term proceeding via the process represented by the subterm y_1 is matched in the $x_1 \parallel (y_1 + y_2)$ term by the possibility of proceeding via the same indeterminate process y_1.

Another similar absorption phenomena is given by the following equation.

$$x_1 \parallel (\alpha y_1 + y_2) \;+\; x_2 \parallel (\alpha y_1 + y_2) \;+\; (x_1 + x_2) \parallel y_2$$
$$+\; (x_1 + x_2) \parallel (\alpha y_1 + y_2)$$
$$= \; x_1 \parallel (\alpha y_1 + y_2) \;+\; x_2 \parallel (\alpha y_1 + y_2)$$
$$+\; \alpha\Big((x_1 + x_2) \parallel y_1\Big) \;+\; (x_1 + x_2) \parallel y_2.$$

Here one of the summands on the right hand side of the equation, $\alpha\Big((x_1 + x_2) \parallel y_1\Big)$, is missing on the left hand side. However, the summand is implicitly there, as the absorbed term, $(x_1 + x_2) \parallel (\alpha y_1 + y_2)$, can be expanded using the Absorption Law (*Abs*) to include the missing summand. Hence we indeed in effect have an absorption law. As before, the summand $(x_1 + x_2) \parallel (\alpha y_1 + y_2)$ is absorbed into the other summands by almost identically the same reasoning as in the previous example.

Indeed these two laws are valid $\sim$-equivalences, as can be verified by a structural induction argument, or by translating them into the left merge language of the previous section, where we have an ω-complete axiomatisation. These two laws do not in themselves suggest any new complexity in our search for an ω-complete axiomatisation for our equivalence. However, as was the case with the

above sequences of reduction laws, these absorption laws come in a whole series of increasingly complex, yet seemingly independent varieties. In fact, we can generate such an absorption law to treat specifically the absorption of any parallel composition of indeterminate processes. For example, consider the term $x_1 \parallel \alpha y_1 \parallel (\beta z_1 + z_2 + z_3)$. The corresponding absorption law tailor-made to absorb this term is given as follows.

$$\begin{aligned}
& x_1 \parallel \alpha y_1 \parallel (\beta z_1 + z_2 + z_3) \\
& \qquad + \ (x_1 + x_2) \parallel \alpha y_1 \parallel (\beta z_1 + z_2 + z_3) \\
& \qquad + \ x_1 \parallel \alpha y_1 \parallel z_2 \ + \ x_1 \parallel \alpha y_1 \parallel z_3 \\
& = \ (x_1 + x_2) \parallel \alpha y_1 \parallel (\beta z_1 + z_2 + z_3) \\
& \qquad + \ \alpha\Big(x_1 \parallel y_1 \parallel (\beta z_1 + z_2 + z_3)\Big) \\
& \qquad + \ \beta(x_1 \parallel \alpha y_1 \parallel z_1) \ + \ x_1 \parallel \alpha y_1 \parallel z_2 \ + \ x_1 \parallel \alpha y_1 \parallel z_3
\end{aligned}$$

Again, as was the case with the reduction laws, these absorption equations can be presented as a single complex axiom schema as follows.

$(Abs_{m,n,p,s_1,t_1,s_2,t_2,\ldots,s_p,t_p})$.

$$\begin{aligned}
& \prod_{i=1}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel \prod_{i=1}^{p} \Big(\sum_{j=1}^{s_i} z_{ij} + \sum_{j=1}^{t_i} \beta_{ij} v_{ij}\Big) \\
& \qquad + \ \sum_{k=1}^{m} \Big[(x_k + x'_k) \parallel \prod_{\substack{i=1 \\ i \neq k}}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel \prod_{i=1}^{p} \Big(\sum_{j=1}^{s_i} z_{ij} + \sum_{j=1}^{t_i} \beta_{ij} v_{ij}\Big)\Big] \\
& \qquad + \ \sum_{k=1}^{p} \sum_{l=1}^{t_k} \Big(\prod_{i=1}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel z_{kl}\Big) \\
& = \ \sum_{k=1}^{m} \Big[(x_k + x'_k) \parallel \prod_{\substack{i=1 \\ i \neq k}}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel \prod_{i=1}^{p} \Big(\sum_{j=1}^{s_i} z_{ij} + \sum_{j=1}^{t_i} \beta_{ij} v_{ij}\Big)\Big] \\
& \qquad + \ \sum_{k=1}^{n} \alpha_k \Big[\prod_{i=1}^{m} x_i \parallel \prod_{\substack{i=1 \\ i \neq k}}^{n} \alpha_i y_i \parallel \prod_{i=1}^{p} \Big(\sum_{j=1}^{s_i} z_{ij} + \sum_{j=1}^{t_i} \beta_{ij} v_{ij}\Big)\Big] \\
& \qquad + \ \sum_{k=1}^{p} \sum_{l=1}^{t_k} \Big(\prod_{i=1}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel z_{kl}\Big) \\
& \qquad + \ \sum_{k=1}^{p} \sum_{l=1}^{s_k} \beta_{kl} \Big(\prod_{i=1}^{m} x_i \parallel \prod_{i=1}^{n} \alpha_i y_i \parallel z_{kl}\Big)
\end{aligned}$$

These sequences of reduction and absorption laws are not completely mutually independent. For instance, given the reduction law

$$(x+v) \parallel (y+z) + x \parallel y + x \parallel z + v \parallel y + v \parallel z$$
$$= x \parallel (y+z) + v \parallel (y+z) + (x+v) \parallel y + (x+v) \parallel z,$$

we could easily deduce the absorption law

$$(x+v) \parallel (y+z) + x \parallel (y+z) + v \parallel (y+z) + (x+v) \parallel y + (x+v) \parallel z$$
$$= x \parallel (y+z) + v \parallel (y+z) + (x+v) \parallel y + (x+v) \parallel z,$$

using the idempotence of $+$. This situation is not surprising, as the reduction laws did set out to provide the least terms to add to a composition in order to eliminate the composition; the corresponding absorption law would necessarily contain at least as much observable behaviour on each side of the equation. On the other hand, we already know that the absorption schema allows us to eliminate parallel composition with simple variable factors in the presence of more complicated terms, which we were not capable of doing with the reduction laws. Hence it appears certain that the absorption laws do add power with which to reason.

There is one important point to note about these sequences. To prove the above laws are valid, we need to invoke structural induction and the Expansion Theorem laws (Exp_{mn}). However, in the above we gave arguments as to why they should be expected to hold valid; namely, every possible single-step behaviour exhibitable by one side of the equation is matched by some single-step behaviour on the other. In the laws which incorporate action symbols explicitly, the Expansion Theorem laws (Exp_{mn}) and the Absorption Law (Abs) are used to simplify the axiom. However, when no action symbols appear explicitly in the axiom, then it is valid by our informal reasoning regardless of the validity of the interleaving Expansion Theorem. That is to say, we can reason that the actionless laws are so basic as to be considered reasonable in *any* notion of equivalence based on behavioural properties of processes. Indeed, some of the above absorption laws were noted in [CAS87] and [CAS88] as valid axioms in their noninterleaving theory of *distributed*

bisimulation. In fact, not only the action-free absorption laws, but also the action-free reduction laws are valid in their theory. We shall exploit the independence of these laws with respect to the nature of any particular behavioural equivalence in **Section 5.3** where we discuss the axiomatisability of any *reasonable* equivalence.

Hence we now find ourselves at a standstill. In searching for an ω-complete set of laws for our semantic congruence over this simple process algebra $\mathcal{P}_2$, we have uncovered a wide range of problems. Starting with the well-known complete theory $\mathcal{T}_2^0$ for closed terms, we discovered we needed to include the following: three straightforward laws for the parallel combinator, (C_1), (C_2) and (C_3), expressing the associativity, commutativity and **0**-absorption of $\|$; an Absorption Law (Abs) motivated by the Expansion Theorem laws (Exp_{mn}); a whole series of reduction laws, $(Red_{t,m_1,n_1,m_2,n_2,\ldots,m_t,n_t})$, describing when a sum of terms involving a large parallel composition could be expressed as a sum of smaller parallel compositions; and finally a whole series of absorption laws, $(Abs_{m,n,p,s_1,t_1,s_2,t_2,\ldots,s_p,t_p})$, describing when a particular parallel composition could be absorbed into another term. Neither of the latter two classes of laws were obvious to discover. Nor is it obvious that no other laws exist which are independent from the above collection. However, defining a normal form for the above theory in order to attempt to prove ω-completeness of this grandiose set of axioms with respect to our semantic congruence is indeed far from trivial.

4.1 Full Merge Language

The language of terms we consider here is the language $\mathcal{P}_2^0$, the set of closed terms over the language $\mathcal{P}_2$ given by the signature $\Sigma_2 = \{\mathbf{0}, ., +, \| \}$. The semantic equivalence which we consider here will once again be the strong observational equivalence $\sim$. We rely on the well-known theory developed for this language and equivalence which tells us that the equivalence is completely characterised by isomorphism between derivation trees, finite unordered trees whose arcs are labelled by elements of **Act**, in which no two identically-labelled arcs lead from the same node to two isomorphic subtrees.

The proof that follows will proceed by induction on the *depth* $|\cdot|$ of terms. Equality throughout this section will represent semantic equality (strong observational equivalence). Thus in our proof, $P = Q$ will mean $P \sim Q$, not necessarily syntactic identity.

The important properties which we shall use are as follows, and are immediate results of the definitions:

$$
\begin{aligned}
&P = Q \text{ implies } |P| = |Q|;\\
&P \neq \mathbf{0} \quad \text{implies} \quad |P \,\|\, Q| > |Q|;\\
&P = Q \text{ and } P \xrightarrow{a} P' \quad \text{implies} \quad Q \xrightarrow{a} Q'\\
&\qquad\qquad \text{for some } Q' = P';\\
&P \xrightarrow{a} P' \quad \text{implies} \quad |P| > |P'|.
\end{aligned}
$$

Definition 4.1.1 *A term P is* irreducible *iff whenever $P = Q \,\|\, R$, we have that either $Q = \mathbf{0}$ or $R = \mathbf{0}$.*

Definition 4.1.2 *A term P is* prime *iff P is irreducible and $P \neq \mathbf{0}$.*

Theorem 4.1.3 (Milner) *Any term $P \in \mathcal{P}_2^0$ can be expressed uniquely (up to $\sim$) as a product (parallel composition) of primes.*

Proof:

That any P can be expressed as a product of primes is straightforward: if $P = \mathbf{0}$, it is equal to the empty product; if P is prime, it is equal to the singleton product, namely itself; otherwise $P = Q \parallel R$ where $Q, R \neq \mathbf{0}$, so by induction on depth, each of Q and R can be expressed as a product of primes:

$$Q = Q_1 \parallel Q_2 \parallel \cdots \parallel Q_m, \qquad R = R_1 \parallel R_2 \parallel \cdots \parallel R_n;$$

Then P can be expressed as a product of primes itself by:

$$P = Q \parallel R = Q_1 \parallel Q_2 \parallel \cdots \parallel Q_m \parallel R_1 \parallel R_2 \parallel \cdots \parallel R_n.$$

The proof presented here that this factorisation is unique proceeds by induction on $|P|$.

Suppose that $P = Q$, but that P and Q have distinct factorisations into products of primes given as follows:

$$P = A_1^{k_1} \parallel A_2^{k_2} \parallel \cdots \parallel A_n^{k_n},$$
$$Q = A_1^{l_1} \parallel A_2^{l_2} \parallel \cdots \parallel A_n^{l_n},$$

where the A_i's are distinct primes (that is, $i \neq j \implies A_i \neq A_j$), and that $k_i, l_i \geq 0$.

Assume that all terms R with $|R| < |P| = |Q|$ have a unique factorisation into a product of primes, and let $\exp(A, R)$ be the exponent of prime A (the number of times A appears) in the unique factorisation of R.

Let m be chosen such that $k_m \neq l_m$, and that $|A_j| > |A_m|$ implies that $k_j = l_j$; that is, A_m is a maximal-sized (wrt depth) prime appearing in the factorisation of P or Q in which the exponents differ. Without loss of generality, we can assume that $k_m > l_m$ (otherwise exchange the roles of P and Q).

The proof proceeds now by cases on the possible form of the factorisation of P:

1) Suppose P is a power of a prime: $P = A_m^{k_m}$;

Firstly, if P is prime (that is, $k_m = 1$), then from $P = Q$, we have that Q is prime, and since $k_m > l_m$, we have that $Q = A_j$ for some $j \neq m$; but then $A_m = A_j$, contradicting the distinctness assumption on the A_i's.

Hence assume that $k_m > 1$;

We can do $P \xrightarrow{a} P'$ for some a, P', and whenever $P \xrightarrow{a} P'$, P' has a unique factorisation with $\exp(A_m, P') = k_m - 1$.

This is true since

$A_m \neq \mathbf{0}$

$\Longrightarrow$ $A_m \xrightarrow{a} R$ *for some* a, R

$\Longrightarrow$ $P \xrightarrow{a} P' = A_m^{k_m - 1} \parallel R$,

and $|P| > |P'| > |R| \Longrightarrow$ *R and P' have unique factorisations given by:*

$$R = A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s,$$

$$P' = A_m^{k_m - 1} \parallel A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s,$$

and $|A'_i| \leq |R| < |A_m| \Longrightarrow A_m \neq A'_i$ *for each i,*

so $\exp(A_m, P') = k_m - 1$.

Suppose that $l_m > 0$;

Then similar to the above, we can do $Q \xrightarrow{a} Q'$ for some a, Q', and have a unique factorisation for Q' in which $\exp(A_m, Q') = l_m - 1$;

But from the above,

$$P \xrightarrow{a} P' \Longrightarrow \exp(A_m, P') = k_m - 1 > l_m - 1,$$

so $P' \neq Q'$;

Therefore $\nexists P' = Q'$ st $P \xrightarrow{a} P'$, contradicting $P = Q$.

Hence assume that $l_m = 0$.

Then from the maximality constraint in the definition of m,

$$l_i > 0 \implies |A_i| \leq |A_m|;$$

Hence whenever $Q \xrightarrow{a} Q'$, Q' *has a unique factorisation in which* $\exp(A_m, Q') = 0$;

This is true since

$Q \xrightarrow{a} Q'$

$\implies A_j \xrightarrow{a} R$ *for some* j, R *st* $l_j > 0$ *where*

$$Q' = A_1^{l_1} \parallel \cdots \parallel A_j^{l_j - 1} \parallel \cdots \parallel A_n^{l_n} \parallel R$$

and R, Q' *have unique factorisations given by:*

$$R = A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s,$$

$$Q' = A_1^{l_1} \parallel \cdots \parallel A_j^{l_j - 1} \parallel \cdots \parallel A_n^{l_n} \parallel A'_1 \parallel \cdots \parallel A'_s,$$

and $|A'_i| \leq |R| < |A_j| \leq |A_m|$ *(as* $l_j > 0$*) for each* i,

so $A'_i \neq A_m$ *for each* i, *so* $\exp(A_m, Q') = l_m = 0$;

But from the above, we can do $P \xrightarrow{a} P'$ *for some* a, P', *and* P' *has a unique factorisation in which* $\exp(A_m, P') = k_m - 1 > 0$;

Therefore $\nexists Q' = P'$ *st* $Q \xrightarrow{a} Q'$, *contradicting* $P = Q$.

2) *Suppose* P *is not a power of a prime:* $\exists j \neq m$ *st* $k_j > 0$;

Let b, T *be such that* $P \xrightarrow{b} T$, *and whenever* $P \xrightarrow{a} P'$, *we have that (since* $|P| > |P'|, |T|$, *and hence* P', T *have unique factorisations)* $\exp(A_m, P') \leq \exp(A_m, T)$.

Then $\exp(A_m, T) \geq k_m$;

This is true since

$A_j \neq 0$

$\implies A_j \xrightarrow{a} R$ *for some* a, R

$\implies P \xrightarrow{a} P' = A_1^{k_1} \parallel \cdots \parallel A_j^{k_j - 1} \parallel \cdots \parallel A_n^{k_n} \parallel R,$

and $|P| > |P'| > |R| \implies R$ *and* P' *have unique factorisations given by:*

$$R = A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s,$$

$$P' = A_1^{k_1} \parallel \cdots \parallel A_j^{k_j - 1} \parallel \cdots \parallel A_n^{k_n} \parallel A'_1 \parallel \cdots \parallel A'_s,$$

so $\exp(A_m, P') \geq k_m$ *(since* $m \neq j$*).*

Let $Q \xrightarrow{b} Q'$;

Then $\exists t, R$ *st* $l_t > 0$ *and* $A_t \xrightarrow{b} R$,

where $Q' = A_1^{l_1} \parallel \cdots \parallel A_t^{l_t - 1} \parallel \cdots \parallel A_n^{l_n} \parallel R$;

Since $|Q| > |Q'| \geq |R|$, R *and* Q' *have unique factorisations given by:*

$$R = A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s,$$

$$Q' = A_1^{l_1} \parallel \cdots A_t^{l_t - 1} \parallel \cdots \parallel A_n^{l_n} \parallel A'_1 \parallel A'_2 \parallel \cdots \parallel A'_s;$$

Suppose that $\exp(A_m, Q') = \exp(A_m, T) \geq k_m > l_m$;

Then $\exists i$ *st* $A'_i = A_m$;

Thus $|A_t| > |R| \geq |A'_i| = |A_m|$, *so by the maximality constraint in the definition of* m, *we have that* $k_t = l_t$, *and* $t \neq m$,

so $\exp(A_m, Q') = l_m + \exp(A_m, R)$;

However now, since $k_t = l_t > 0$, *we also have that*

$$P \xrightarrow{b} P' = A_1^{k_1} \parallel \cdots \parallel A_t^{k_t - 1} \parallel \cdots \parallel A_n^{k_n} \parallel R,$$

and

$$\begin{aligned} \exp(A_m, P') &= k_m + \exp(A_m, R) \\ &> l_m + \exp(A_m, R) = \exp(A_m, Q'); \end{aligned}$$

Hence $\exp(A_m, T) \geq \exp(A_m, P') > \exp(A_m, Q')$, *so* $Q' \neq T$;

Therefore $\nexists Q' = T$ *st* $Q \xrightarrow{b} Q'$, *contradicting* $P = Q$. □

We state one important corollary of this result here.

Corollary 4.1.4 (Simplification Lemma) *For* P, Q *and* $R \in \mathcal{P}_2^0$,

$$P \parallel R = Q \parallel R \text{ implies } P = Q.$$

Proof:

Let P, Q *and* R *have unique factorisations given by*

$$\begin{aligned} P &= P_1 \parallel P_2 \parallel \cdots \parallel P_m \\ Q &= Q_1 \parallel Q_2 \parallel \cdots \parallel Q_n \\ R &= R_1 \parallel R_2 \parallel \cdots \parallel R_t \end{aligned}$$

Then clearly the unique factorisations for $P \parallel R$ *and* $Q \parallel R$ *must be given by*

$$P \parallel R = P_1 \parallel P_2 \parallel \cdots \parallel P_m \parallel R_1 \parallel R_2 \parallel \cdots \parallel R_t$$
$$Q \parallel R = Q_1 \parallel Q_2 \parallel \cdots \parallel Q_n \parallel R_1 \parallel R_2 \parallel \cdots \parallel R_t$$

But since $P \parallel R = Q \parallel R$, *these factorisations must be identical.*

Hence P *and* Q *must themselves have identical prime factors, and so* $P = Q$. □

4.2 A Simpler Proof

In this section we present a much simplified proof of the above factorisation theorem. The proof derives from the above simplification lemma, which is first proven independent of the unique factorisation theorem.

Lemma 4.2.1 (Simplification Lemma) *For* P, Q *and* $R \in \mathcal{P}_2^0$,

$$P \parallel R = Q \parallel R \quad \textit{implies} \quad P = Q.$$

Proof:

We actually prove the following two results by simultaneous induction on $|P| + |Q| + |R|$:

(i) $P \parallel R = Q \parallel R$ *implies* $P = Q$;

(ii) $R \xrightarrow{a} R'$ *and* $P \parallel R = Q \parallel R'$ *implies* $Q \xrightarrow{a} Q'$ *for some* $Q' = P$.

(i) Let $P \parallel R = Q \parallel R$, *and* $P \xrightarrow{a} P'$;

Then $P \parallel R \xrightarrow{a} P' \parallel R$, *so* $\exists S = P' \parallel R$ *st* $Q \parallel R \xrightarrow{a} S$;

Hence either

(a) $\exists Q'$ *st* $Q \xrightarrow{a} Q'$ *and* $Q' \parallel R = P' \parallel R$, *or*

(b) $\exists R'$ *st* $R \xrightarrow{a} R'$ *and* $Q \parallel R' = P' \parallel R$;

For (a), by induction hypothesis (i), $Q' = P'$;

For (b), by induction hypothesis (ii), $\exists Q' = P'$ *st* $Q \xrightarrow{a} Q'$.

Similarly, $P \parallel R = Q \parallel R$ *and* $Q \xrightarrow{a} Q'$

implies $\exists P' = Q'$ *st* $P \xrightarrow{a} P'$.

Hence $P \parallel R = Q \parallel R$ *implies* $P = Q$.

(ii) Let $R \xrightarrow{a} R'$, *and* $P \parallel R = Q \parallel R'$;

Then $P \parallel R \xrightarrow{a} P \parallel R'$, *so* $\exists S = P \parallel R'$ *st* $Q \parallel R' \xrightarrow{a} S$;

Hence either

(a) $\exists Q'$ *st* $Q \xrightarrow{a} Q'$ *and* $Q' \parallel R' = P \parallel R'$; *or*

(b) $\exists R''$ *st* $R' \xrightarrow{a} R''$ *and* $Q \parallel R'' = P \parallel R'$;

For (a), by induction hypothesis (i), $Q' = P$;

For (b), by induction hypothesis (ii), $\exists Q' = P$ *st* $Q \xrightarrow{a} Q'$.

Hence in any case, $\exists Q' = P$ *st* $Q \xrightarrow{a} Q'$. □

Our result now follows quite simply.

Theorem 4.2.2 (Unique Factorisation of Processes) *Any term* $P \in \mathcal{P}_2^0$ *can be expressed uniquely (up to* $\sim$*) as a product of primes.*

Proof:

We shall not repeat the argument that the prime decomposition exists, but rather just argue uniqueness. This we shall do by induction on $|P|$.

Thus suppose first that $P = Q$, *and that* P *and* Q *have prime factorisations given by*

$$\begin{aligned} P &= C \parallel A_1 \parallel A_2 \parallel \cdots \parallel A_k, \\ Q &= C \parallel B_1 \parallel B_2 \parallel \cdots \parallel B_l; \end{aligned}$$

That is, the two factorisations have a common prime factor.

Then by the **Simplification Lemma 4.2.1**, *we have*

$$A_1 \parallel A_2 \parallel \cdots A_k \;=\; B_1 \parallel B_2 \parallel \cdots \parallel B_l;$$

By the inductive hypothesis, $A_1 \parallel A_2 \parallel \cdots \parallel A_k$ *and* $B_1 \parallel B_2 \parallel \cdots B_l$ *must be identical prime factor decompositions;*

Thus the prime factor decompositions for P *and* Q *above are identical.*

Hence suppose that $P = A_1 \parallel A_2 \parallel \cdots A_k$ *and* $Q = B_1 \parallel B_2 \parallel \cdots \parallel B_l$ *are prime factor decompositions such that for all* i *and* j, $A_i \neq B_j$;

If $k = 1$ *or* $l = 1$, *then* $P = Q$ *is prime, so* $k = l = 1$, *and* $A_1 = B_1$, *contradicting the distinctness of the* A_i*'s and* B_j*'s;*

Hence assume that $k, l \geq 2$;

Assume further that for all i *and* j, $|A_1| \leq |A_i|, |B_j|$;

Let a, R *be such that* $A_1 \xrightarrow{a} R$, *and let* R *have a unique factorisation (as* $|R| < |A_1| \leq |P|$*) given by*

$$R \;=\; R_1 \parallel R_2 \parallel \cdots \parallel R_r;$$

Then $P \xrightarrow{a} P'$ *with unique prime factorisation*

$$P' \;=\; R_1 \parallel R_2 \parallel \cdots \parallel R_r \parallel A_2 \parallel \cdots \parallel A_k;$$

Thus $Q \xrightarrow{a} Q' = P'$, *so some* $B_j \xrightarrow{a} T$ *with*

$$Q' \;=\; B_1 \parallel \cdots \parallel B_{j-1} \parallel T \parallel B_{j+1} \parallel \cdots \parallel B_l;$$

Assume that $j = 1$, *and that* T *has a unique prime factorisation given by*

$$T \;=\; T_1 \parallel T_2 \parallel \cdots \parallel T_t;$$

Then $Q' = P'$ *have unique factorisations given by*

$$R_1 \parallel \cdots \parallel R_r \parallel A_2 \parallel \cdots \parallel A_k \;=\; T_1 \parallel \cdots \parallel T_t \parallel B_2 \parallel \cdots \parallel B_l;$$

Now these are identical prime factorisations;

But for all i and j, $A_i \neq B_j$,

and for all i and j, $|R_i| \leq |R| < |A_1| \leq |B_j|$,

so $R_i \neq B_j$;

Thus no term B_j appears in the factorisation

$$R_1 \parallel \cdots \parallel R_r \parallel A_2 \parallel \cdots \parallel A_k,$$

so $l < 2$, contradicting the assumption that $k, l \geq 2$. □

4.3 Adding Communication

The theorem of the previous section is valid when we use the merge with communication operator $|$ instead of the merge-only interleaving operator $\parallel$. The proof is similar to the original proof of the previous result, but we must be careful to recognise the possibility of two processes communicating to allow a τ transition.

The setup to the theorem is identical to the last section, except for the language which we consider. Here we are taking the language $\mathcal{P}_3^0$, the set of closed terms over the language $\mathcal{P}_3$ given by the signature $\Sigma_3 = \{\mathbf{0}, ., +, |\}$. The semantic equivalence remains as the strong observational congruence $\sim$, and we can still characterise the equivalence using derivation trees. Once again, equality throughout will represent semantic equality (strong observational equivalence). The definitions of *irreducible* and *prime* are the same as before.

We shall use $a, b, \ldots \in \Lambda \subseteq \mathbf{Act}$ to range over the non-τ actions (that is, $\tau \notin \Lambda$), and $\mu, \nu, \ldots \in \mathbf{Act} = \Lambda \cup \{\tau\}$ to range over *all* atomic action.

Theorem 4.3.1 (Unique Factorisation of Processes) *Any term $P \in \mathcal{P}_3^0$ can be expressed uniquely (up to $\sim$) as a product (parallel composition) of primes.*

Proof:

Again, that any P can be expressed as a product of primes is straightforward. The proof that this factorisation is unique again proceeds by induction on $|P|$, and the cases we consider are the same, only more care must be taken in each.

Suppose that $P = Q$, but that P and Q have distinct factorisations into products of primes given as follows:

$$P = A_1^{k_1} \mid A_2^{k_2} \mid \cdots \mid A_n^{k_n}$$
$$Q = A_1^{l_1} \mid A_2^{l_2} \mid \cdots \mid A_n^{l_n},$$

where the A_i's are distinct primes (that is, $i \neq j \implies A_i \neq A_j$), and that $k_i, l_i \geq 0$.

Assume that all terms R with $|R| < |P| = |Q|$ have a unique factorisation into a product of primes, and again let $\exp(A, R)$ be the exponent of prime A (the number of times A appears as a factor) in the unique factorisation of R.

Let m be chosen such that $k_m \neq l_m$, and that $|A_j| > |A_m|$ implies that $k_j = l_j$; that is, A_m is a maximal-sized (wrt depth) prime appearing in the factorisation of P or Q in which the exponents differ. Without loss of generality, we can assume that $k_m > l_m$ (otherwise exchange the roles of P and Q).

The proof proceeds now by cases on the possible form of the factorisation of P:

1) *Suppose P is a power of a prime: $P = A_m^{k_m}$;*

   Firstly, if P is prime (that is, $k_m = 1$), then from $P = Q$, we have that Q is prime, and since $k_m > l_m$, we have that $Q = A_j$ for some $j \neq m$; but then $A_m = A_j$, contradicting the distinctness assumption on the A_i's.

   Hence assume that $k_m > 1$;

   Suppose that $l_m > 0$;

If $A_m \xrightarrow{a} A'_m$ *for some* $a \in \Lambda$,

then $Q \xrightarrow{a} Q'$ *with* $\exp(A_m, Q') = l_m - 1$;

But $P \xrightarrow{a} P'$

$\Longrightarrow \exp(A_m, P') = k_m - 1 > l_m - 1$;

Hence $\nexists P' = Q'$ *st* $P \xrightarrow{a} P'$,

contradicting $P = Q$.

If $A_m \xrightarrow{\mu} A'_m \Longrightarrow \mu = \tau$,

then $P \xrightarrow{\tau} P' \Longrightarrow \exp(A_m, P') = k_m - 1$ *still;*

and still $Q \xrightarrow{\tau} Q'$

with $\exp(A_m, Q') = l_m - 1 < k_m - 1$;

So again $\nexists P' = Q'$ *st* $P \xrightarrow{\tau} P'$,

contradicting $P = Q$.

Hence assume that $l_m = 0$.

Then from the maximality constraint in the definition of m,

$l_i > 0 \Longrightarrow |A_i| \leq |A_m|$;

Hence $Q \xrightarrow{\mu} Q' \Longrightarrow Q'$ *has a unique factorisation in which*

$\exp(A_m, Q') = 0$;

But we can do $P \xrightarrow{\mu} P'$ *for some* μ, P'

such that $\exp(A_m, P') = k_m - 1 > 0$

Therefore $\nexists Q' = P'$ *st* $Q \xrightarrow{\mu} Q'$, *contradicting* $P = Q$.

2) Suppose $\exists j \neq m$ *st* $k_j > 0$;

Let μ, T *be such that* $P \xrightarrow{\mu} T$, $|P| = |T| + 1$, *and whenever* $P \xrightarrow{\nu} P'$, *with* $|P| = |P'| + 1$, *we have that (since* $|P| < |P'|, |T|$, *and hence* P', T *have unique factorisations)*

$\exp(A_m, P') \leq \exp(A_m, T)$.

Then clearly $\exp(A_m, T) \geq k_m$.

Suppose $Q \xrightarrow{\mu} Q' = T$;

Then we have one of three cases:

i) $\exists s, R$ *st* $l_s > 0$ *and* $A_s \xrightarrow{\mu} R$,

where $Q' = A_1^{l_1} \mid \cdots \mid A_s^{l_s - 1} \mid \cdots \mid A_n^{l_n} \mid R$;

Hence $|A_s| = |R| + 1$;

If $|A_s| \leq |A_m|$,

then $\exp(A_m, Q') \leq l_m < k_m \leq \exp(A_m, T)$,

(contradicting $Q' = T$*).*

If $|A_s| > |A_m|$,

then $\exp(A_m, Q') = l_m + \exp(A_m, R)$,

and $s \neq m$, *and* $k_s = l_s > 0$;

So $P \xrightarrow{\mu} P' = A_1^{k_1} \mid \cdots \mid A_s^{k_s - 1} \mid \cdots \mid A_n^{k_n} \mid R$

with $\exp(A_m, P') = k_m + \exp(A_m, R)$;

and $|P| = |P'| + 1$;

Thus $\exp(A_m, Q') < \exp(A_m, P') \leq \exp(A_m, T)$;

(contradicting $Q' = T$*).*

ii) $\mu = \tau$ *and* $\exists s, t, R_s, R_t, a$ *st* $l_s, l_t > 0$

and $A_s \xrightarrow{a} R_s$ *and* $A_t \xrightarrow{\bar{a}} R_t$,

where $Q' = A_1^{l_1} \mid \cdots \mid A_s^{l_s - 1} \mid \cdots$

$\cdots \mid A_t^{l_t - 1} \mid \cdots \mid A_n^{l_n} \mid R_s \mid R_t$;

But then $|Q'| \leq |Q| - 2 = |P| - 2 < |T|$

(contradicting $Q' = T$*).*

iii) $\mu = \tau$ *and* $\exists s, R, R', a$ *st* $l_s > 1$

and $A_s \xrightarrow{a} R$ *and* $A_s \xrightarrow{\bar{a}} R'$,

where $Q' = A_1^{l_1} \mid \cdots \mid A_s^{l_s - 2} \mid \cdots$

$\cdots \mid A_n^{l_n} \mid R \mid R$;

But then again $|Q'| \leq |Q| - 2 = |P| - 2 < |T|$

(contradicting $Q' = T$*).*

Therefore $\nexists Q' = T$ *st* $Q \xrightarrow{\mu} Q$, *contradicting* $P = Q$. □

Again we close this section by stating the same important corollary.

Corollary 4.3.2 (Simplification Lemma) *For* P, Q *and* $R \in \mathcal{P}_3^0$,

$$P \mid R = Q \mid R \quad \textit{implies} \quad P = Q.$$

Proof:

As in **Corollary 4.1.4.** □

Notice that the Prime Decomposition Theorem is again a rather simple corollary of the Simplification Lemma. However in this case we would find it quite difficult to prove the Simplification Lemma without recourse to the Prime Decomposition Theorem.

4.4 Adding Silent τ's

In this section, we shall once again prove a unique factorisation theorem for our language $\mathcal{P}_3^0$, but this time under (weak) observational congruence $\approx^c$. In this case, we face problems with the definitions we have been using. We cannot proceed blindly, as the theorem fails immediately. For instance, consider the term $\tau.\mathbf{0}$; this is not prime, as

$$\tau.\mathbf{0} \approx^c \tau.\mathbf{0} \mid \tau.\mathbf{0}.$$

But there is no decomposition of the term $\tau.\mathbf{0}$ into prime factors.

The above problem arises everywhere in which a τ-prefix appears, due to the following proposition.

Proposition 4.4.1 $\tau.P \approx^c \tau.\mathbf{0} \mid P$.

Proof:

By structural induction on P, using the alternate definition of $\approx^c$ given by **Proposition 2.2.9.** □

Hence it would appear that no process term of the (semantic) form $\tau.P$ can be expressed as a product of prime factors. To remedy the situation here, we work only with what Milner calls *proper normal form* terms, and we rely on the the accompanying theory from *e.g.,* [MIL85].

Definition 4.4.2 *A term* $\sum_{i=1}^{n} \mu_i P_i$ *is in* proper normal form *(*pnf*) iff*

(i) It does not take the form $\tau.P$*;*

(ii) Each P_i *is in proper normal form;*

(iii) For $k \neq j$*, no* μ_k*-derivative of* $\mu_j.P_j$ *is sumcongruent to* P_k*.*

In this definition, a *μ-derivative* of a term P is defined to be any term P' such that $P \stackrel{\mu}{\Longrightarrow} P'$; and two terms are said to be *sumcongruent* precisely when they can be proven to be equal using only axioms (A_1) and (A_2) from the previous chapter, the associativity and commutativity of the nondeterministic choice operator $+$, along with the usual laws of equational reasoning. An important property of sumcongruence with respect to proper normal form terms is given by the following proposition.

Proposition 4.4.3 *For proper normal form terms* P *and* Q*,* $P \approx Q$ *implies* P *and* Q *are sumcongruent, and so in particular,* $P \approx^c Q$*.*

Proof:

See Case 1 of the proof to **Theorem 3.1** *of* [MIL85]. □

What we shall show is that any proper normal form term has a unique factorisation into a parallel product of primes. With this we shall be as close as possible to a complete decomposition theorem as we could possibly get, using the following propositions.

Proposition 4.4.4 *If* $P = \sum_{i=1}^{n} \mu_i.p_i$ *is in pnf, then* $P \not\approx^c \tau.P'$ *for any* P'*.*

Proof:

Firstly, if $P \approx^c \tau.P'$*, then* $P \approx^c \tau.P$*, as*

$$P \approx^c \tau.P' \approx^c \tau.\tau.P' \approx^c \tau.P.$$

Thus suppose $P \approx^c \tau.P$;

Then $P \stackrel{\tau}{\Longrightarrow} p'$ *for some* $p' \approx P$;

Hence for some j *with* $1 \leq j \leq n$, $\mu_j = \tau$ *and* $p_j \Longrightarrow p'$;

By part (ii) of **Definition 4.4.2**, p' *must itself be in pnf;*

Hence from **Proposition 4.4.3**, $P \approx^c p'$;

If $n = 1$, *then* $P = \tau.p_1$, *contradicting part (i) of* **Definition 4.4.2**;

Hence $n > 1$, *and there exists* $k \neq j$ *with* $1 \leq k \leq n$;

Now $P \stackrel{\mu_k}{\longrightarrow} p_k$, *so* $p' \stackrel{\mu_k}{\Longrightarrow} p'' \approx p_k$, *and again* p'' *and* p_k *are in fact sumcongruent;*

But then p'' *is a* μ_k*-derivative of* $\mu_j.p_j$ *which is sumcongruent to* p_k, *contradicting part (iii) of* **Definition 4.4.2.** □

Proposition 4.4.5 (Normal Form Lemma) *Any term* $P \in \mathcal{P}_3^0$ *is congruent to either a proper normal form term or a term* $\tau.P'$, *where* P' *is in proper normal form.*

Proof:

See [MIL85], **Section 3.3.** □

Proposition 4.4.6 (Hennessy) *For terms* $P, Q \in \mathcal{P}_3^0$, $P \approx Q$ *iff either:*

(i) $P \approx^c Q$; *or*

(ii) $P \approx^c \tau.Q$; *or*

(iii) $\tau.P \approx^c Q$.

Proof:

($\Longleftarrow$) *Straightforward, using* $P \approx \tau.P$;

($\Longrightarrow$) *By* **Proposition 4.4.5**, *there are pnf terms* P' *and* Q' *such that either* $P \approx^c P'$ *or* $P \approx^c \tau.P'$, *and* $Q \approx^c Q'$ *or* $Q \approx^c \tau.Q'$;

In any case, $P \approx Q$ *implies* $P' \approx Q'$, *so by* **Proposition 4.4.3**, $P' \approx^c Q'$;

Then depending on the cases above, one of the the three conclusions in the proposition must hold. □

Thus given any term P, either $P \not\approx^c \tau.P$, in which case we will be able to express it as a proper normal form term, and hence by the result to follow as a unique parallel composition of primes; or else $P \approx^c \tau.P$, in which case by **Proposition 4.4.5**, we can express P as $P \approx^c \tau.P'$ for some P' in proper normal form, so by **Proposition 4.4.1**, $P \approx^c \tau.\mathbf{0} \mid P'$, and so we will be able to express P uniquely as a parallel composition of primes (corresponding to the factorisation of P'), composed in parallel with the term $\tau.\mathbf{0}$.

When restricting our attention to proper normal form terms, we do not run up against the difficulty in the mismatch between equivalence $\approx$ and congruence $\approx^c$ in derivations of congruent terms as given in the alternate definition of $\approx^c$ given by **Proposition 2.2.9**. This is due to **Proposition 4.4.3**, which tells us that $\approx$ and $\approx^c$ coincide on the subset of proper normal form terms. Hence for proper normal form terms, the definition of our congruence $\approx^c$ is exactly the largest (weak) bisimulation relation defined over the set of proper normal form terms as given in **Definition 2.2.8**. More importantly for us in what follows, the mismatch is equally remedied in the alternate definition of observational congruence given by **Proposition 2.2.9**.

Furthermore, syntactic depth of terms is preserved by congruence over proper normal form terms, as congruent pnf terms are sumcongruent, and laws (A_1) and (A_2), as well as the laws of equational logic, respect syntactic depth. Also, proper normal form terms are closed under transition derivations, by part *(ii)* of **Definition 4.4.2**. Finally, any factors of a pnf term are themselves expressible as pnf terms, as

$$P \approx^c \tau.P \implies P \mid Q \approx^c \tau.(P \mid Q).$$

This result is equally valid in the reverse direction, that is, if all factors of a parallel composition are expressible as pnf terms, then the composition itself is expressible as a pnf term. This reverse result is demonstrated by the following propositions.

Proposition 4.4.7 *For pnf terms P, Q and R, if $P \approx Q \mid R$ then $|P| = |Q| + |R|$.*

Proof:

By induction on $|Q| + |R|$.

Firstly, if $P \approx \mathbf{0} \approx Q \mid R$,

then $Q, R \approx \mathbf{0}$, so $|P| = |Q| = |R| = 0$;
(as P, Q, R and $\mathbf{0}$ are sumcongruent)

Hence $|P| = |Q| + |R|$.

Also, if $Q \approx \mathbf{0}$,

then $|Q| = 0$ and $P \approx R$, so $|P| = |R| = |Q| + |R|$.
(as P and R are sumcongruent)

Therefore, assume $P, Q \not\approx \mathbf{0}$;

Let μ and P' be such that $P \xrightarrow{\mu} P'$ and $|P| = 1 + |P'|$;

Then $Q \mid R \stackrel{s}{\Longrightarrow} Q' \mid R' \approx P'$ for some $s \in \mathbf{Act}^*$ st $\hat{s} = \hat{\mu}$;

If $s = \varepsilon$, then $\mu = \tau$ and $P' \approx Q \mid R \approx P$;

But then since P and P' are sumcongruent, $|P| = |P'|$
(contradiction)

Hence $s \neq \varepsilon$, so $|Q'| + |R'| < |Q| + |R|$;

Thus by the inductive hypothesis, $|P'| = |Q'| + |R'| < |Q| + |R|$;

So $|P| \leq |Q| + |R|$;

Now let μ and Q' be such that $Q \xrightarrow{\mu} Q'$ and $|Q| = 1 + |Q'|$;

Then $P \stackrel{\mu}{\Longrightarrow} P' \approx Q' \mid R$, *so by the inductive hypothesis,*

$$|P'| = |Q'| + |R| = |Q| + |R| - 1$$

Thus $|P| \geq 1 + |P'| \geq |Q| + |R|$;

Therefore we have that $|Q| + |R| \leq |P| \leq |Q| + |R|$,

so $|P| = |Q| + |R|$. □

Proposition 4.4.8 *For pnf terms* P, Q *and* R, *if* $P \approx Q \mid R$ *then* $P \approx^c Q \mid R$. *Therefore in particular, if* Q *and* R *can be expressed as pnf terms, then* $Q \mid R$ *can be expressed as a pnf term as well.*

Proof:

$P \not\approx^c \tau.(Q \mid R)$; *(by* **Proposition 4.4.4***)*

Suppose $\tau.P \approx^c Q \mid R$;

Then $Q \mid R \stackrel{\tau}{\Longrightarrow} Q' \mid R' \approx P$,

so $|P| = |Q'| + |R'| < |Q| + |R| = |P|$;

(by **Proposition 4.4.7***)*

(contradiction);

Hence by **Proposition 4.4.6**, $P \approx^c Q \mid R$. □

These propositions then easily generalise giving the following proposition.

Proposition 4.4.9 *For pnf terms* P *and* P_i *(for* $1 \leq i \leq n$*), if* $P \approx \prod_{i=1}^{n} P_i$, *then* $|P| = \sum_{i=1}^{n} |P_i|$, *and* $P \approx^c \prod_{i=1}^{n} P_i$.

Proof:

By induction on n, *using the previous two propositions.* □

Thus we can easily check that we indeed have the following desired properties for our proof (where P, Q and R are parallel compositions of pnf's):

$$P \approx^c Q \text{ implies } |P| = |Q|;$$
$$P \not\approx^c \mathbf{0} \text{ implies } |P \mid Q| > |Q|;$$
$$P \approx^c Q \text{ and } P \xrightarrow{\mu} P' \text{ implies } \exists Q' \approx^c P' \text{ such that } Q \stackrel{\mu}{\Longrightarrow} Q';$$
$$P \xrightarrow{\mu} P' \text{ implies } |P| > |P'|.$$

The proof of the unique factorisation theorem that follows will proceed by induction on the *depth* $|\cdot|$ of terms. Again, equality throughout the proof will represent semantic equality (observational congruence). Thus in our proof, $P = Q$ will mean $P \approx^c Q$, not necessarily syntactic identity.

Theorem 4.4.10 (Unique Factorisation of Processes) *Any term $P \in \mathcal{P}_3^0$ in proper normal form can be expressed uniquely (up to $\approx^c$) as a product (parallel composition) of primes (in proper normal form).*

Proof:

Again, that any pnf term P can be expressed as a product of primes is straightforward. If $P = \mathbf{0}$ or P is prime, then P is equal to the empty product or singleton product, respectively. If $P = Q \mid R$ where $Q, R \neq \mathbf{0}$, then we can assume Q and R to be pnf terms with depths strictly less than that of P, so by induction, Q and R can be expressed as products of primes; taking the product of these products expresses P itself as a product of primes.

The proof that this factorisation is unique again proceeds by induction on $|P|$, and the cases we consider are the same as before, only again much more care must be taken in each.

Suppose that $P = Q$, but that P and Q have distinct factorisations into products of primes given as follows:

$$P = A_1^{k_1} \mid A_2^{k_2} \mid \cdots \mid A_n^{k_n}$$
$$Q = A_1^{l_1} \mid A_2^{l_2} \mid \cdots \mid A_n^{l_n},$$

where the A_i's are distinct primes (that is, $i \neq j$ implies $A_i \neq A_j$), and that $k_i, l_i \geq 0$.

Assume that every pnf term R with $|R| < |P| = |Q|$ has a unique factorisation into a product of primes, and let $\exp(A, R)$ be the exponent of prime A (the number of times A appears) in the unique factorisation of R.

Let m be such that $k_m \neq l_m$, and that whenever $|A_j| > |A_m|$ we have that $k_j = l_j$; that is, A_m is a maximal-sized (wrt depth) prime appearing in the factorisation of P or Q in which the exponents differ. Without loss of generality, we can assume that $k_m > l_m$ (otherwise exchange the roles of P and Q).

The proof proceeds now by cases on the possible form of the factorisation of P:

1) *Suppose P is a power of a prime: $P = A_m^{k_m}$;*

   Firstly, if P is prime (that is, $k_m = 1$), then from $P = Q$, we have that Q is prime, and since $k_m > l_m$, we have that $Q = A_j$ for some $j \neq m$; but then $A_m = A_j$, contradicting the distinctness assumption on the A_i's.

   Hence assume that $k_m > 1$;

   Suppose that $l_m = 0$;

   Then from the maximality constraint in the definition of m,

   $$l_i > 0 \text{ implies } |A_i| \leq |A_m|;$$

   Hence $Q \stackrel{\mu}{\Longrightarrow} Q'$ implies $\exp(A_m, Q') = 0$;

   But for some μ, R, $A_m \xrightarrow{\mu} R$,

   $$\text{so } P \xrightarrow{\mu} P' \text{ with } \exp(A_m, P') = k_m - 1 > 0;$$

   Therefore $\nexists Q' = P'$ st $Q \stackrel{\mu}{\Longrightarrow} Q'$, contradicting $P = Q$.

   Hence assume that $l_m > 0$;

   Let μ, T be such that $A_m \xrightarrow{\mu} T$, and $|A_m| = |T| + 1$;

Then $Q \xrightarrow{\mu} Q'$ *with* $\exp(A_m, Q') = l_m - 1$, *and* $|Q'| = |Q| - 1$;

Suppose that $P \stackrel{\mu}{\Longrightarrow} P' = Q'$;

Then $|P'| = |P| - 1$, *so clearly* $P' \equiv A_m^{k_m - 1} \mid A'_m$

for some A'_m *st* $A_m \xrightarrow{\mu} A'_m$;

Hence $\exp(A_m, P') = k_m - 1 > l_m - 1 = \exp(A_m, Q')$,

so $P' \neq Q'$;

Therefore $\nexists P' = Q'$ *st* $P \stackrel{\mu}{\Longrightarrow} P'$, *contradicting* $P = Q$.

2) *Suppose* $\exists j \neq m$ *st* $k_j > 0$;

Let μ, T *be such that* $P \xrightarrow{\mu} T$, $|P| = |T| + 1$, *and whenever*

$P \xrightarrow{\nu} P'$ *with* $|P| = |P'|+1$, *we have (since* $|P| < |P'|, |T|$,

and hence P', T *have unique factorisations)*

$\exp(A_m, P') \leq \exp(A_m, T)$.

Then clearly $\exp(A_m, T) \geq k_m$.

Suppose $Q \stackrel{\mu}{\Longrightarrow} Q' = T$;

Then $|Q'| = |Q| - 1$, *so clearly* $\exists t, R$ *st* $l_t > 0$ *and* $A_t \xrightarrow{\mu} R$,

where $Q' = A_1^{l_1} \mid \cdots \mid A_t^{l_t - 1} \mid \cdots \mid A_n^{l_n} \mid R$;

and $|A_t| = |R| + 1$;

If $|A_t| \leq |A_m|$,

then $\exp(A_m, Q') \leq l_m < k_m \leq \exp(A_m, T)$,

(contradicting $Q' = T$*).*

If $|A_t| > |A_m|$,

then $\exp(A_m, Q') = l_m + \exp(A_m, R)$,

and $t \neq m$, *and* $k_t = l_t > 0$;

So $P \xrightarrow{\mu} P' = A_1^{k_1} \mid \cdots \mid A_t^{k_t - 1} \mid \cdots \mid A_n^{k_n} \mid R$

with $\exp(A_m, P') = k_m + \exp(A_m, R)$;

and $|P| = |P'| + 1$;

Thus $\exp(A_m, Q') < \exp(A_m, P') \leq \exp(A_m, T)$;

(contradicting $Q' = T$*).*

Therefore $\nexists Q' = T$ *st* $Q \xrightarrow{\mu} Q'$, *contradicting* $P = Q$. □

Corollary 4.4.11 *Any term $P \in \mathcal{P}_3^0$ can be expressed uniquely either as the parallel product of primes, or as the parallel product of primes in parallel with the process $\tau.\mathbf{0}$.*

Proof:

By the remarks preceding the proof of the above theorem, any term P such that $P \neq \tau.P$ can be expressed uniquely as the parallel composition of primes which are proper normal form terms. Clearly these are primes over the whole language $\mathcal{P}_3^0$, and conversely, primes in $\mathcal{P}_3^0$ have proper normal forms. Hence any P such that $P \neq \tau.P$ can be expressed uniquely as a parallel product of primes.

For any term P such that $P = \tau.P$, we can find a proper normal form term P' such that $P = \tau.P' = \tau.\mathbf{0} \mid P'$. Thus we can express P uniquely as the process $\tau.\mathbf{0}$ parallelly-composed with the parallel composition of primes. □

Finally we again state the same important corollary.

Corollary 4.4.12 (Simplification Lemma) *For P, Q and $R \in \mathcal{P}_3^0$,*

$$P \mid R \approx Q \mid R \quad \textit{implies} \quad P \approx Q.$$

Proof:

Similar to **Corollary 4.1.4.** □

Notice that because of the problems introduced by the silent τ action, we cannot state this corollary any stronger. For instance, $a \mid \tau = \tau a \mid \tau$, but $a \neq \tau a$; we can only infer here that $a \approx \tau a$, not $a \approx^c \tau a$.

However, if we restricted the terms P, Q and R in the Simplification Lemma to be parallel compositions of pnf terms, then the lemma would be stated as usual, with equality (semantic congruence) rather than equivalence in the premise and conclusion.

Chapter 5

Nonexistence of Finite Axiomatisations

In this chapter we deal with problems of proving the nonexistence of finite axiomatisations in various process algebras involving a symmetric parallel combinator. Initially, we demonstrate the non-finite-axiomatisability of the Expansion Theorem in the sublanguage $\mathcal{P}_2^0$ of closed terms of the language containing the full merge operator in its signature, with respect to strong observational congruence. That is to say, we show that no finite set of axioms will suffice for an equational theory to completely characterise strong observational congruence of our full merge language. The proof will make no further assumptions on the set $\mathbf{Act}$ of atomic actions other than it being non-empty; we simply assume that there exists some $a \in \mathbf{Act}$. Thus the proof will hold even for the most restrictive case where there is only one distinguishable atomic action. We then note how the proof extends easily to the case where we allow communication as well as merge.

Next we extend the first result to problems in the axiomatisation of stricter noninterleaving semantic congruences. We shall show in fact that any *reasonable* congruence defined over our full merge language $\mathcal{P}_2^0$ cannot be finitely axiomatised, where we define a congruence to be reasonable in a rigorous manner. The proof presented in this final section will in fact cover the case of strong observational congruence, and so subsume the result in the first section. However, the former proof is included in the text as a stepping stone towards the more complicated proof of the latter result.

The proofs to follow will be very much proof-theoretic (as opposed to model-theoretic) in nature. We shall be considering the possible natural deduction style proof trees of certain valid statements, and shall often be making observations about the forms of terms appearing in the trees. For this purpose, we shall find ourselves making heavy use of the unique decomposition properties of the previous chapter in order to restrict the possible syntactic forms which may appear in the trees.

5.1 Saturated Axiomatisations

The equivalences to which we shall restrict ourselves will all respect **0**-absorption through both the $+$ and $\|$ operators, and in the sequel we shall want to deal exclusively with terms which do not contain any unnecessary **0** summands or factors. With this in mind, we define $\tilde{t}$ to be the term t with all **0** summands and factors removed. Formally we have the following definition:

$$\tilde{\mathbf{0}} = \mathbf{0} \qquad \widetilde{t+u} = \begin{cases} \tilde{t} & \text{if } |u| = 0 \wedge fv(u) = \emptyset \\ \tilde{u} & \text{if } |t| = 0 \wedge fv(t) = \emptyset \\ \tilde{t} + \tilde{u} & \text{otherwise} \end{cases}$$

$$\tilde{x} = x$$

$$\widetilde{at} = a\tilde{t} \qquad \widetilde{t \| u} = \begin{cases} \tilde{t} & \text{if } |u| = 0 \wedge fv(u) = \emptyset \\ \tilde{u} & \text{if } |t| = 0 \wedge fv(t) = \emptyset \\ \tilde{t} \,\|\, \tilde{u} & \text{otherwise} \end{cases}$$

We shall also restrict the type of axiom set which we shall allow in our proof system, to exploit the above **0** absorption properties in our proofs. The special class of axiomatisations will allow us to prove statements without invoking unnecessary **0** factors and summands. However, as we shall see, the restricted class will not be a real restriction with respect to the properties of axiomatisability which we are analysing. That is, given any arbitrary finite, sound and complete axiomatisation, we can produce another finite, sound and complete axiomatisation which is in our special class of axiom sets.

The axiom sets to which we shall restrict ourselves will be *saturated*, as defined as follows.

Definition 5.1.1 *Let $\mathcal{T}$ be an arbitrary set of equational axioms. The* saturation *of $\mathcal{T}$ is defined to be*

$$Sat(\mathcal{T}) \;=\; \mathcal{T} \cup \widetilde{\mathcal{T}},$$

where

$$\widetilde{\mathcal{T}} \;=\; \Big\{ \tilde{t}_0 = \tilde{u}_0 \;\Big|\; \exists t, u, \bar{x} \subseteq fv(t) \cup fv(u) \; st \; t = u \;\in\; \mathcal{T}, \\ t_0 = t\Big\{\overline{0}/_{\bar{x}}\Big\}, \;\; and \;\; u_0 = u\Big\{\overline{0}/_{\bar{x}}\Big\}\Big\}.$$

Proposition 5.1.2 *$Sat(\mathcal{T}) = Sat(Sat(\mathcal{T}))$.*

Proof:

Immediate from the definition of $Sat(\mathcal{T})$. □

Proposition 5.1.3 *$\mathcal{T} \vdash t = u$ if and only if $Sat(\mathcal{T}) \vdash t = u$.*

Proof:

Again immediate from the definition of $Sat(\mathcal{T})$. □

Proposition 5.1.4 *$\mathcal{T}$ is finite if and only if $Sat(\mathcal{T})$ is finite.*

Proof:

Again immediate from the definition of $Sat(\mathcal{T})$. □

Thus from now on, we shall restrict ourselves to considering only saturated axiom sets, that is, axiom sets $\mathcal{T}$ such that $\mathcal{T} = Sat(\mathcal{T})$. As we pointed out earlier, the above results show that this assumption is not a restriction if we are interested in finite, sound and complete axioms sets. However, an important simplification of proofs is given as follows.

Proposition 5.1.5 *If we have a proof of a statement $P = Q$ in our natural deduction style proof system parameterised by a saturated axiom set $\mathcal{T}$, then replacing $p = q$ throughout the proof tree by $\tilde{p} = \tilde{q}$ gives us a valid proof of the statement $\tilde{P} = \tilde{Q}$. Thus using a saturated axiom set, a (shortest) proof of a result containing no occurrences of* $\mathbf{0}$ *as a summand or as a factor need not contain any occurrence of* $\mathbf{0}$ *as a summand or factor in any of its intermediate terms.*

Proof:

It is not hard to see that any inference:

$$\frac{\cdots\ p_i = q_i\ \cdots}{p = q}(rule)$$

can be changed to a valid inference:

$$\frac{\cdots\ \tilde{p}_i = \tilde{q}_i\ \cdots}{\tilde{p} = \tilde{q}}(rule').$$

The only nontrivial case is in dealing with axioms; here we have:

$$\frac{}{p = q}(t = u)$$

where $p = q$ is axiom $t = u$ instantiated by some substitution σ. This inference can be replaced by:

$$\frac{}{\tilde{p} = \tilde{q}}(\tilde{t}_0 = \tilde{u}_0)$$

where

$$t_0 = t\left\{\overline{\mathbf{0}}/_{\bar{x}}\right\} \text{ and } u_0 = u\left\{\overline{\mathbf{0}}/_{\bar{x}}\right\},$$

$$\text{where } \bar{x} = \{x \mid \sigma_x = \mathbf{0}\};$$

Clearly, $\tilde{p} = \tilde{q}$ is axiom $\tilde{t_0} = \tilde{u_0}$ instantiated with substitution $\tilde{\sigma}$

(i.e., $\sigma_x = r \implies \tilde{\sigma}_x = \tilde{r}$). □

Thus we restrict our proof system to be as described in **Section 2.3.1**, parameterised by saturated axiom sets.

5.2 Strong Congruence

The process language we are considering here is the language $\mathcal{P}_2^0$, which is the set of closed terms in the language $\mathcal{P}_2$ given by the signature $\Sigma_2 = \{\mathbf{0}, ., +, \| \}$. The semantic equivalence we are considering is again the strong observational congruence $\sim$. As we saw, this congruence is completely characterised by the theory $\mathcal{T}_2^0$ consisting of the following (infinite) set of laws:

$$(A_1)\ \ (x+y)+z = x+(y+z) \qquad (A_3)\ \ x+x = x$$
$$(A_2)\ \ x+y = y+x \qquad (A_4)\ \ x+\mathbf{0} = x$$

$$(Exp_{mn})\ \textit{For}\ P = \sum_{i=1}^{m} \alpha_i P_i\ \textit{and}\ Q = \sum_{j=1}^{n} \beta_j Q_j,$$

$$P \| Q = \sum_{i=1}^{m} \alpha_i (P_i \| Q) + \sum_{j=1}^{n} \beta_j (P \| Q_j)$$

In the sequel, we shall use $=$ to represent $\sim$ (semantic equality), and $\equiv$ to represent syntactic identity modulo associativity and commutativity of the operators $+$ and $\|$. For ease of presentation, we shall also extend the transition system $\longrightarrow$ to allow $P \xrightarrow{a} R$ whenever $\exists P' = R$ such that $P \xrightarrow{a} P'$.

We proceed first to present several technical results which we shall rely on in our proof of the nonexistence of a finite equational axiomatisation of our equivalence.

Proposition 5.2.1 $P = Q \implies |P| = |Q|$ *and* $\Omega(P) = \Omega(Q)$.

Proof:

This follows easily from invariance through the laws of equational logic, and through our axioms. □

Proposition 5.2.2

(i) $P = \mathbf{0} \iff |P| = 0 \iff \Omega(P) = 0$;

(ii) $P = a$ *for some* $a \in \mathbf{Act} \iff |P| = 1$.

Proof:

These follow by structural induction on P. □

Proposition 5.2.3 *If* P *is reducible, then* $\Omega(P) > 1$.

Proof:

If $P = Q \parallel R$ *where* $Q, R \neq \mathbf{0}$,

then $\Omega(Q), \Omega(R) \geq 1$,

so $\Omega(P) = \Omega(Q) + \Omega(R) \geq 2$. □

Definition 5.2.4 *Let* $a \in \mathbf{Act}$ *be fixed (note that this only requires that* $\mathbf{Act} \neq \emptyset$*, which we have said would be our only assumption on* **Act***). Then let* $\mathcal{A}_n$ *and* φ_n $(n \geq 0)$ *be defined as follows:*

$$\mathcal{A}_0 \stackrel{\text{def}}{=} \mathbf{0};$$

$$\mathcal{A}_{n+1} \stackrel{\text{def}}{=} a\mathcal{A}_n \quad (n > 0);$$

$$\varphi_n \stackrel{\text{def}}{=} \sum_{i=1}^{n} \mathcal{A}_i \quad (n \geq 0).$$

The proof of our main result in this section will rely heavily on special properties of these sequences of process terms. These important properties are as presented in the following sequence of propositions.

Proposition 5.2.5 *For all* $m, n \geq 0$ *with* $m \neq n$ *we have that* $\mathcal{A}_m \neq \mathcal{A}_n$*, and for all* $m, n > 1$ *we have that* $\mathcal{A}_m \neq \varphi_n$.

Proof:

$$m \neq n \implies |\mathcal{A}_m| = m \neq n = |\mathcal{A}_n|$$
$$\implies \mathcal{A}_m \neq \mathcal{A}_n.$$

$\Omega(\varphi_n) = 1 \neq n = |\varphi_n|$ *(for* $n > 1$*),*

but $|A_m| = m = \Omega(A_m)$,

so for all $m, n > 1$, $\mathcal{A}_m \neq \varphi_n$. □

Proposition 5.2.6 *φ_n is prime for each $n > 0$.*

Proof:

This is easily seen to be true, as $\Omega(\varphi_n) = 1$. □

Proposition 5.2.7 $P \parallel Q = \mathcal{A}_n$ iff $\exists i, j$ *st* $P = \mathcal{A}_i$, $Q = \mathcal{A}_j$, *and* $i + j = n$.

Proof:

($\Longleftarrow$) $\mathcal{A}_i \parallel \mathcal{A}_j = \mathcal{A}_{i+j}$ *by induction on* $i + j$;

($\Longrightarrow$) *a is prime (as $\Omega(a) = 1$), so the unique factorisation of $\mathcal{A}_m$ into primes is given by*

$$\mathcal{A}_m = \underbrace{a \parallel a \parallel \cdots \parallel a}_{m}.$$

Thus if $P \parallel Q = \mathcal{A}_m$, and P and Q have unique factorisations given by

$$P = \Psi_1 \parallel \Psi_2 \parallel \cdots \parallel \Psi_i \quad \text{and} \quad Q = \Psi_{i+1} \parallel \Psi_{i+2} \parallel \cdots \parallel \Psi_n,$$

then each $\Psi_i = a$, and so $P = \mathcal{A}_i$ and $Q = \mathcal{A}_{n-i}$, and $n = m$.

□

5.2.1 Preliminary Results

In this section we state and prove the technical lemmata which we need to derive our main result in the following section. Firstly however, we define a few propositions on pairs of sets of terms which will designate properties of equations which we want to analyse in our proof system.

Definition 5.2.8 *For $n > 1$ and $U, V \subseteq \mathcal{P}_2^0$ being two sets of terms, let $\Theta_n^L(U,V)$ be the proposition which states the following:*

$$P \in U \cup V \implies P \equiv \tilde{P}, \quad \text{and} \quad P \not\equiv \mathbf{0}, P' + P'',$$
$$\text{and} \quad a \parallel \varphi_n = \sum U = \sum V,$$
$$\text{and} \quad \exists P \in U \text{ st } P = a \parallel \varphi_n,$$
$$\text{and} \quad \nexists Q \in V \text{ st } Q = a \parallel \varphi_n.$$

Thus $\Theta_n^L(U,V)$ states (among other things) that the equation $\sum U = \sum V$ expresses a (valid) equality between terms equal to $a \parallel \varphi_n$ in which the term $a \parallel \varphi_n$ is already captured by a single summand on the left hand side of the equality, but not by any single summand on the right hand side.

Then let $\Theta_n(U,V) = \Theta_n^L(U,V) \vee \Theta_n^L(V,U)$.

Proposition 5.2.9 *Let $n > 1$ and $U, V \subseteq \mathcal{P}_2^0$ be such that $\Theta_n(U,V)$, and let $P \in U \cup V$ be the term satisfying $P = a \parallel \varphi_n$; Then $P \equiv A \parallel P_n$, where $A = a$ and $P_n = \varphi_n$.*

Proof:

$a \parallel \varphi_n \xrightarrow{a} \varphi_n$ and $a \parallel \varphi_n \xrightarrow{a} \mathcal{A}_k$ for each $k : 1 \leq k \leq n$;
Hence $P \not\equiv aP'$ as $aP' \xrightarrow{a} P'$ only,
but φ_n, $\mathcal{A}_k$ $(1 \leq k \leq n)$ are all distinct;
Thus $P \equiv P' \parallel P''$ where $P', P'' \neq \mathbf{0}$;

Suppose the unique prime factorisations of P' and P'' are given by

$$P' = P_1' \parallel P_2' \parallel \cdots \parallel P_k' \quad \text{and} \quad P'' = P_1'' \parallel P_2'' \parallel \cdots \parallel P_l'';$$

Then since a and φ_n are prime, we have that for some primes $A = a$ and $P_n = \varphi_n$,

$$P \equiv P' \parallel P'' = P_1' \parallel P_2' \parallel \cdots \parallel P_k' \parallel P_1'' \parallel P_2'' \parallel \cdots \parallel P_l''$$
$$\equiv A \parallel P_n,$$

so $k = l = 1$, $P' = P'_1$ *and* $P'' = P''_1$,

and either $P'_1 = a$ *and* $P''_1 = \varphi_n$, *or* $P'_1 = \varphi_n$ *and* $P''_1 = a$;

Hence either $P' = a$ *and* $P'' = \varphi_n$, *or* $P' = \varphi_n$ *and* $P'' = a$;

Thus $P \equiv A \parallel P_n$, *where* $A = a$ *and* $P_n = \varphi_n$. □

The following proposition is the main technical result on which the nonexistence of a finite axiomatisation proof rests, and has a correspondingly lengthy proof.

Proposition 5.2.10 *Let* $\mathcal{F}$ *be a finite saturated set of sound (with respect to strong observational congruence* $\sim$*) axioms, and let* n *be bigger than the number of operators in any axiom of* $\mathcal{F}$. *Then no axiom* $t = u$ *in* $\mathcal{F}$ *can be instantiated to a statement* $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *such that* $\Theta_n(U, V)$.

Proof:

Let n *be as above, and suppose* $t = u$ *is an axiom in* $\mathcal{F}$ *such that under substitution* σ, $t = u$ *instantiates to* $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *such that* $\Theta_n(U, V)$.

Without loss of generality, assume that $\Theta_n^L(U, V)$;

Clearly, $fv(t) = fv(u)$, *as* $t = u$ *is assumed to be a valid axiom,*

and if $x \in fv(t) \setminus fv(u)$, *then choosing* $M > |u|$ *and defining substitution* σ *by*

$$\sigma(x) = \begin{cases} \mathbf{0}, & \text{if } x \in fv(u), \\ \mathcal{A}_M, & \text{otherwise,} \end{cases}$$

we would have that

$$|t\sigma| \geq M > |u\sigma|,$$

so that $t\sigma \not\sim u\sigma$, *and hence that* $t \not\sim u$.

$t \equiv t_1 + t_2 + \cdots + t_k$ *and* $u \equiv u_1 + u_2 + \cdots + u_{k'}$ *for some* $k, k' > 0$,

where each $t_i, u_i \not\equiv v + v'$;

$\Theta_n^L(U,V) \implies$ *for some* i, *either* $t_i\sigma \equiv A \parallel P_n$ *or* $t_i\sigma \equiv A \parallel P_n + Q$,

where $A = a$ *and* $P_n = \varphi_n$;

Consider the structure of t_i:

$t_i \equiv \mathbf{0} \implies t_i\sigma \equiv \mathbf{0}$ (contradiction);

$t_i \equiv x \implies \sigma_x \equiv t_i\sigma$ *and* $x \in fv(u_j)$ *for some* j

$\implies u_j \not\equiv \mathbf{0}, au', u' + u'', u' \parallel u''$

$\implies u_j \equiv x$ *and* $A \parallel P_n \in V$

(contradicting $\Theta_n^L(U,V)$);

$t_i \equiv at' \implies t_i\sigma \equiv a(t'\sigma)$ (contradiction);

$t_i \equiv t' + t'' \implies$ (contradiction);

Thus $t_i \equiv t' \parallel t''$ *and* $t_i\sigma \equiv t'\sigma \parallel t''\sigma = a \parallel \varphi_n$;

Hence $t_i \equiv t' \parallel t''$ *with* $t'\sigma \equiv A = a$ *and* $t''\sigma \equiv P_n = \varphi_n$;

Now $t'' \equiv v_1 + v_2 + \cdots v_l$ *where* $l < n$ *and each* $v_h \not\equiv v + v'$;

$t''\sigma \equiv v_1\sigma + v_2\sigma + \cdots v_l\sigma = \varphi_n = \mathcal{A}_1 + \mathcal{A}_2 + \cdots + \mathcal{A}_n$,

so some $v_h\sigma = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$ *for some* $m > 1$ *and*

$0 < r_1 < r_2 < \cdots < r_m$;

Thus clearly $v_h \not\equiv \mathbf{0}, av, v + v', v \parallel v'$, *so* $v_h \equiv x$ *for some variable* x *where* $\sigma_x = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$;

Clearly $x \notin fv(t')$, *as* $|t'\sigma| = 1 < r_m = |\sigma_x|$;

Let $\sigma' = \sigma\{a\varphi_n/x\}$;

Then $t'\sigma' \equiv t'\sigma$, *and* $t\sigma' \xrightarrow{a} t'\sigma' \parallel \varphi_n = a \parallel \varphi_n$;

Therefore for some j, $u_j\sigma' \xrightarrow{a} a \parallel \varphi_n$;

Now $|u_j\sigma'| > n + 1 = |u\sigma|$, *so clearly* $x \in fv(u_j)$;

Consider the structure of u_j:

$u_j \equiv \mathbf{0} \implies x \notin fv(u_j)$ (contradiction);

$u_j \equiv x \implies u_j\sigma' \equiv a\varphi_n \not\xrightarrow{a} a \parallel \varphi_n$ (contradiction);

$u_j \equiv au' \implies u_j\sigma' \equiv a(u'\sigma')$

$\implies u'\sigma' = a \parallel \varphi_n$ *and* $x \in fv(u')$

$\implies u' \equiv \omega_1 + \omega_2 + \cdots + \omega_l$ *for some* l *with each*

$\omega_h \not\equiv \omega + \omega'$, *and* $x \in fv(\omega_m)$ *for some* m;

Consider the structure of the ω_m *with* $x \in fv(\omega_m)$:

$\omega_m \equiv \mathbf{0} \implies x \notin fv(\omega_m)$

(contradiction);

$\omega_m \equiv a\omega \implies x \in fv(\omega)$

$\implies n+1 \leq |\omega\sigma'| < |\omega_m\sigma'|$

$\leq |u'\sigma'| = n+1$

(contradiction);

$\omega_m \equiv \omega + \omega' \implies$ (contradiction);

$\omega_m \equiv \omega \parallel \omega' \implies x \in fv(\omega)$ *or* $x \in fv(\omega')$

$\implies n+1 < |\omega\sigma'| + |\omega'\sigma'| = |\omega_m\sigma'|$

$\leq |u'\sigma'| < n+1$

(contradiction);

Thus $x \in fv(\omega_m) \implies \omega_m \equiv x$ *and* $\omega_m\sigma' = a\varphi_n$;

But $u'\sigma' \equiv \sum_{m=1}^{l} \omega_m\sigma' = a \parallel \varphi_n$

$= a\varphi_n + \mathcal{A}_2 + \mathcal{A}_3 + \cdots + \mathcal{A}_{n+1}$;

So $\sum_{\substack{m=1 \\ x \notin fv(\omega_m)}}^{l} \omega_m\sigma + a\varphi_n = a \parallel \varphi_n$,

or $\sum_{\substack{m=1 \\ x \notin fv(\omega_m)}}^{l} \omega_m\sigma = \mathcal{A}_2 + \mathcal{A}_3 + \cdots + \mathcal{A}_{n+1}$;

Thus $n+1 \leq |u'\sigma| < |u\sigma| = n+1$ (contradiction);

$u_j \equiv u' + u'' \implies$ (contradiction);

Hence $u_j \equiv u' \parallel u''$ *with* $x \in fv(u')$;

Now since $u''\sigma \xrightarrow{a} p$ *for some* p, *we have* $u\sigma \xrightarrow{a} u'\sigma \parallel p$;

Thus $u'\sigma \parallel p = \varphi_n$ *or* $u'\sigma \parallel p = \mathcal{A}_r$ *for some* $r : 1 < r < n$;

$$u'\sigma \parallel p = \varphi_n \implies p = 0 \text{ and } u'\sigma = \varphi_n$$

(since $u'\sigma \neq 0$, and φ_n is prime)

$$\implies u_j\sigma \equiv u'\sigma \parallel u''\sigma = a \parallel \varphi_n$$

(contradicting $\Theta_n^L(U,V)$)

$$u'\sigma \parallel p = \mathcal{A}_r \implies u'\sigma = \mathcal{A}_{r'} \text{ for some } r' \leq r;$$

But $x \in fv(u')$, and $\sigma_x = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$;

Hence clearly $u'\sigma \neq \mathcal{A}_{r'}$ for any r' (contradiction)

Therefore no axiom $t = u$ in $\mathcal{F}$ can be instantiated to a statement $p = q$ where $p \equiv \sum U$ and $q \equiv \sum V$ such that $\Theta_n(U,V)$. □

Hence we have that the axioms alone cannot generate arbitrarily complex valid equations of the form we are analysing. The following proposition further restricts the possible ways of generating these statements as resulting leaves of proof trees. With these results, our main non-finite-axiomatisability result will follow quite immediately.

Proposition 5.2.11 *Suppose in a sound proof, we have an inference:*

$$\frac{p = r \quad r = q}{p = q}(trans)$$

where $p \equiv \sum U$, $q \equiv \sum V$, $r \equiv \sum W$,

and $R \in W \implies R \equiv \tilde{R}$, and $R \not\equiv 0$, $R' + R''$;

Then

$$\Theta_n(U,V) \implies \Theta_n(U,W) \vee \Theta_n(W,V).$$

Similarly for the (sub_+) rule; corresponding to the inference:

$$\frac{p = q \quad p' = q'}{p + p' = q + q'}(sub_+)$$

where $p \equiv \sum U$, $q \equiv \sum V$, $p' \equiv \sum U'$, and $q' \equiv \sum V'$, we have the result that

$$\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U, V) \vee \Theta_n(U', V').$$

Therefore, a statement $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *for some* U *and* V *satisfying* $\Theta_n(U, V)$ *cannot be initially introduced into a proof tree as the result of the application of either the* (*trans*) *rule nor the* (sub_+) *rule.*

Proof:

Consider the (*trans*) *rule case:*

Assume $\Theta_n^L(U, V)$*; We know immediately that*

$$P \in U \cup V \cup W \implies P \equiv \tilde{P} \text{ and } P \not\equiv \mathbf{0},\ P' + P'',$$

and (from $\Theta_n^L(U, V)$*, and the soundness of the proof in which the inference appears) that*

$$a \parallel \varphi_n = \sum U = \sum V = \sum W;$$

Now if $\nexists R \in W$ *st* $R = a \parallel \varphi_n$*, then clearly* $\Theta_n^L(U, W)$*;*
And if $\exists R \in W$ *st* $R = a \parallel \varphi_n$*, then clearly* $\Theta_n^L(W, V)$*;*
Similarly, $\Theta_n^L(V, U) \implies \Theta_n^L(W, U) \vee \Theta_n^L(V, W)$*;*
Hence $\Theta_n(U, V) \implies \Theta_n(U, W) \vee \Theta_n(W, V)$.

The (sub_+) *rule case is similarly straightforward:*

Assume $\Theta_n^L(U \cup U', V \cup V')$*; Again we know immediately that*

$$P \in U \cup U' \cup V \cup V' \implies P \equiv \tilde{P} \text{ and } P \not\equiv \mathbf{0},\ P' + P'',$$

and that

$$\exists P \in U \cup U' \text{ st } P = a \parallel \varphi_n;$$

Suppose this $P \in U$*; then (from* $\Theta_n^L(U \cup U', V \cup V')$*, and the soundness of the proof in which the inference appears) we have that*

$$a \parallel \varphi_n = \sum U = \sum V,$$

so clearly $\Theta_n^L(U, V)$;

And similarly, if this $P \in U'$, *then* $\Theta_n^L(U', V')$.

Similarly, $\Theta_n^L(V \cup V', U \cup U') \implies \Theta_n^L(V, U) \vee \Theta_n^L(V', U')$;

Hence $\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U, V) \vee \Theta_n(U', V')$. □

5.2.2 Main Result

Here we state and prove our main theorem, the nonexistence of a finite axiomatisation for our congruence.

Theorem 5.2.12 *Let* $\mathcal{F}$ *be a finite saturated set of sound (with respect to strong observational congruence* $\sim$*) axioms, and let* n *be large enough (as allowed by* **Proposition 5.2.10***) so that no axiom in* $\mathcal{F}$ *can be instantiated to express any truth* $p = q$ *where*

$$p \equiv \sum U \text{ and } q \equiv \sum V \text{ such that } \Theta_n(U, V).$$

Then our system cannot prove the valid statement

$$a \parallel \varphi_n = a\varphi_n + \mathcal{A}_2 + \mathcal{A}_3 + \cdots + \mathcal{A}_{n+1}.$$

Hence no finite complete axiom system can exist for strong congruence $\sim$.

Proof:

Suppose we have a (shortest) proof of the statement

$$a \parallel \varphi_n = a\varphi_n + \mathcal{A}_2 + \mathcal{A}_3 + \cdots + \mathcal{A}_{n+1}$$

which involves no terms containing **0** *as a summand or a factor. The proof takes the following form:*

$$\frac{\mathcal{D}_0}{p = q}(rule),$$

where $p \equiv \sum U_0$ *and* $q \equiv \sum V_0$ *for*

$$U_0 = \{a \parallel \varphi_n\} \quad \text{and} \quad V_0 = \{a\varphi_n, \mathcal{A}_2, \mathcal{A}_3, \cdots, \mathcal{A}_{n+1}\},$$

so clearly $\Theta_n(U_0, V_0)$ *holds.*

Since this must be a finite *proof, somewhere in the proof tree is an inference*

$$\frac{\mathcal{D}}{\sum U = \sum V}(rule) \qquad \text{where} \quad \Theta_n(U, V),$$

such that the premise $\mathcal{D}$ *of the inference contains no equality*

$$\sum U' = \sum V' \quad \text{where} \quad \Theta_n(U', V');$$

By **Proposition 5.2.11**, $(rule)$ *can be neither of* $(trans)$ *nor* (sub_+)*;*

Furthermore, by **Proposition 5.2.10**, *we know that* $(rule)$ *cannot be* $(t = u)$ *for any axiom* $t = u \in \mathcal{F}$*;*

Also clearly $(rule)$ *cannot be* $(symm)$*, as* $\Theta_n(U, V) \iff \Theta_n(V, U)$*;*

Finally, $(rule)$ *cannot be any of* $(refl)$*,* (sub_α)*, or* $(sub_\parallel)$*, as this would contradict* $\Theta_n(U, V)$*;*

Hence we have shown that the original statement cannot be proven.

□

5.2.3 Adding Communication

We could repeat the above proof for the nonexistence of a finite axiomatisation for strong congruence over the language $\mathcal{P}_3$ which contains the parallel combinator which allows communication (synchronisation of complimentary actions) as well as merging of actions. The proof would be identical under the assumption that our atomic action a was not its own complimentary action (that is, $\bar{a} \neq a$). The reason that the proof would remain unchanged is that a is the only action symbol which appears in any term used throughout the proof, so no communication could

occur, and the communicating case would degenerate to the non-communicating full merge case. Hence we do not bother repeating the argument here.

5.3 Noninterleaving Semantic Congruences

In this section, we shall in one fell swoop prove that any *"reasonable"* congruence which is at least as discriminating as observational congruence is not finitely axiomatisable over the language $\mathcal{P}_2^0$ whose signature contains the symmetric full merge operator but not the left merge operator. In particular, we shall show that any attempt at axiomatising such a "reasonable" notion of noninterleaving semantic equivalence is doomed to suffer the pitfalls of non-finite-axiomatisability. The only hope for such systems is either to go outside of the system to introduce new operators, for example, to incorporate the left merge operator $\lfloor\!\lfloor$, or else to find some "nice" axiom schemata in the spirit of the Expansion Theorem.

The observations on which we build here are those made in **Section 3.3** while working with the language $\mathcal{P}_2$, looking for an ω-complete axiom set. There we discovered an abundance of unexpected arbitrarily-complex independent axioms which had to be included in a complete set of axioms. Some of the same observations in a different line of study — that of axiomatising a certain noninterleaving semantic equivalence, *distributed bisimulation equivalence* — were made in [CAS87]. Using the insight gained from those observations, we shall here modify the proof of this chapter to apply not just to observational congruence, but to any reasonable congruence which is at least as discriminating.

We must state precisely what we mean for a congruence $\cong$ over $\mathcal{P}_2^0$ to be "reasonable". Much thought has gone into the problem of settling the question of exactly what identities should hold in a good semantic equivalence. Firstly, as stated already, the equivalence should definitely be a congruence, thus allowing the validity of substitutivity of program parts. Secondly, there are very strong arguments that any terms which are identified should at least be observationally congruent; the arguments which distinguish observationally distinct processes

should hold valid for our hypothetical "reasonable" equivalence. Notice here however that we are not abstracting away from internal events which should not be observable to the environment.

Outside of this, there is little agreement as to how fine a congruence should be. The greatest arguments stem from the Petri net community and other proponents of *noninterleaving semantics.* The objections to the grossity of observational congruence arise due to its property, introduced by the Expansion Theorem, of identifying terms involving distinct causal dependencies on their actions. For instance, a simple application of the Expansion Theorem would quickly lead us to conclude that

$$a \parallel b = ab + ba.$$

However, whereas on the left hand side of this statement, there is no causal dependencies expressed between the two actions a and b — the two actions are simply performed independently — the summands of the term on the right hand side each express a definite causal relationship between the actions; in the first summand, action a must occur before action b, whereas this situation is reversed in the second summand. Such an interleaving semantic understanding of processes reduces parallelism to a nonprimitive operation definable in terms of nondeterministic choice and causal dependency. Objections arising against this viewpoint stem from the belief that parallelism should not be expressed as above, but rather that it has properties which should guarantee it a place among the set of primitive concepts.

Thus we contend that the reasonable congruence which we are seeking is strictly finer than observational congruence. But how far must we cut down on this equivalence in order to reach a "reasonable" congruence? We want to cut down on it far enough to avoid all possible objections to the treatment of concurrency. For instance, we would not want to allow the following partial application of the Expansion Theorem:

$$a \parallel b = a \parallel b + ab.$$

This could possibly be considered a valid law in some noninterleaving semantic theory, as the concurrent nature of the atomic processes are still present on both sides of the equation. However, it still allows the introduction of causal dependency where it had not previously existed, and as such is faced with the same arguments faced originally by the Expansion Theorem.

However, we do not want to allow our "reasonable" congruence to be too fine. For instance, Winskel's *event structure semantics* ([WIN83]), as well as the original event structure semantics of Boudol and Castellani (**Section 3** of [BOU86]) only allow process terms to be identified if they are identical modulo the associativity, commutativity and **0**-absorption of the $+$ and $\|$ combinators, as well as the associativity of a sequential combinator in the latter case. Clearly these approaches are too strict, as they do not allow for any non-trivial identities, not even the well-accepted idempotence of $+$.

What we in fact argue here is that some of the reduction laws introduced in **Section 3.3** are acceptable identities to make in any reasonable congruence. For instance we want to allow the following identity:

$$\begin{aligned}(x+v)\,\|\,(y+z) \;+\; x\,\|\,y \;+\; x\,\|\,z \;+\; v\,\|\,y \;+\; v\,\|\,z \\ = \;\; x\,\|\,(y+z) \;+\; v\,\|\,(y+z) \;+\; (x+v)\,\|\,y \;+\; (x+v)\,\|\,z.\end{aligned}$$

As argued in **Section 3.3**, this reduction law can be informally justified as follows: every possible single-step behaviour which one side of the equation can exhibit is matched by an identical single step behaviour on the other side of the equation within an identical parallel context. For example, the possibility of the indeterminate process x proceeding in the second summand $x\,\|\,y$ on the left hand side of the equation is matched by the possibility of the same indeterminate process x proceeding in the third summand $(x+v)\,\|\,y$ on the right hand side of the equation: both allow the indeterminate process x to proceed in the context where it is running in parallel with the indeterminate process y. This reduction law, as with many of the other reduction and absorption laws introduced in **Section 3.3**, does not introduce causal dependency where it did not previously exist. Indeed, it does

not mention any action terms explicitly. Thus it is not open to the objections faced by the Expansion Theorem.

Hence, to summarise, we want our "reasonable" congruence to be at least as fine as observational congruence, and to satisfy reduction phenomena of the above form. That is, we would like any congruence to satisfy the following sequence of reduction laws.

$$\Big(\sum_{i=1}^{m} x_i\Big) \parallel \Big(\sum_{j=1}^{n} y_j\Big) \;+\; \sum_{i=1}^{m}\sum_{j=1}^{n} (x_i \parallel y_j)$$
$$= \sum_{i=1}^{m} \Big[x_i \parallel \Big(\sum_{j=1}^{n} y_j\Big)\Big] \;+\; \sum_{j=1}^{n} \Big[\Big(\sum_{i=1}^{m} x_i\Big) \parallel y_j\Big].$$

Notice that this is just a small subset of the laws from **Section 3.3** which we could argue to be reasonable. However, we only need to consider this sequence to prove our result. Thus by taking $m = 2$ in the above schema, and allowing the substitutions $x_i, y_i := \mathcal{A}_i$, we want to consider it to be reasonable that our congruence satisfy the following sequence of reduction laws (one for each $n \geq 0$):

$$(Red_n) \quad \varphi_2 \parallel \varphi_n \;+\; \sum_{i=1}^{2}\sum_{j=1}^{n} \mathcal{A}_i \parallel \mathcal{A}_j \;=\; \sum_{i=1}^{2} \mathcal{A}_i \parallel \varphi_n \;+\; \sum_{j=1}^{n} \varphi_2 \parallel \mathcal{A}_j.$$

For instance, for $n = 3$ we have the law (Red_3) given as follows:

$$\begin{aligned}
&(a + aa) \parallel (a + aa + aaa) \\
&\qquad + a \parallel a \;+\; a \parallel aa \;+\; a \parallel aaa \\
&\qquad + aa \parallel a \;+\; aa \parallel aa \;+\; aa \parallel aaa \\
&= a \parallel (a + aa + aaa) \;+\; aa \parallel (a + aa + aaa) \\
&\qquad + (a + aa) \parallel a \;+\; (a + aa) \parallel aa \;+\; (a + aa) \parallel aaa.
\end{aligned}$$

Recall now that our proof of the nonexistence of a finite axiomatisation for strong observational congruence $\sim$ proceeded as follows: We demonstrated a certain set of equivalences $\{s_i \sim t_i \mid i \geq 0\}$ such that given any finite set F of $\sim$-valid axioms, we could choose n big enough so that $F \not\vdash s_n = t_n$.

Suppose we consider now a new congruence $\cong$ which is stronger (finer) than observational congruence $\sim$. Suppose we also have that $\forall i \geq 0 : s_i \cong t_i$. Then as a corollary of the above result, we would have that $\cong$ was not finitely axiomatisable, for if it were (say some set F finitely axiomatised $\cong$), then we would have that $F \vdash s_n = t_n \ \ \forall n \geq 0$. As $\cong \subseteq \sim$, F would be a finite set of $\sim$-valid laws, so our property that $F \not\vdash s_n = t_n$ for some n would contradict the completeness of F for the congruence $\cong$.

Thus to extend this proof to our wider class of equivalences, we need to replace the set $\{s_i = t_i \mid i \geq 0\}$ of $\sim$-equivalences with one containing $\cong$-equivalences. We will in fact use the sequence of reduction laws Red_n mentioned above for this purpose, thus allowing our proof to apply to any "reasonable" equivalence.

5.3.1 Preliminary Results

In this section we make some technical definitions and state and prove the technical lemmata which we need to derive our main result in the following section. We shall continue in this section to use $=$ to represent strong congruence $\sim$ and $\equiv$ to represent syntactic identity modulo associativity and commutativity of the operators $+$ and $\|$.

Firstly, we want to restrict our attention to a certain subset of process terms as defined as follows.

Definition 5.3.1 *For any arbitrary integer $n > 2$, we define $\vec{S}_n$ to be the derivation and a-prefix of derivation closure of the set $\{\varphi_2 \| \varphi_n\}$. That is, $\vec{S}_n$ is the smallest set satisfying:*

$$i) \ \ \varphi_2 \| \varphi_n \in \vec{S}_n;$$
$$ii) \ \ P \in \vec{S}_n,\ P \xrightarrow{a} P' \implies P', aP' \in \vec{S}_n.$$

We can express this set explicitly as follows.

Proposition 5.3.2

$$\begin{aligned}\vec{\mathcal{S}}_n = &\left\{\varphi_2 \parallel \varphi_n\right\} \cup \left\{\mathcal{A}_i \parallel \mathcal{A}_j \;\middle|\; 0 \le i \le 2,\ 0 \le j \le n\right\} \\ &\cup \left\{\mathcal{A}_i \parallel \varphi_n \;\middle|\; 0 \le i \le 2\right\} \cup \left\{\varphi_2 \parallel \mathcal{A}_j \;\middle|\; 0 \le j \le n\right\} \\ &\cup \left\{a(\mathcal{A}_i \parallel \varphi_n) \;\middle|\; 0 \le i < 2\right\} \cup \left\{a(\varphi_2 \parallel \mathcal{A}_j) \;\middle|\; 0 \le j < n\right\} \\ &\cup \left\{a(\mathcal{A}_i \parallel \mathcal{A}_j) \;\middle|\; 0 \le i \le 2,\ 0 \le j \le n,\ i+j \le n+1\right\}.\end{aligned}$$

Proof:

Straightforward. □

Proposition 5.3.3 *If $P + Q = \sum S$ for some $S \subseteq \vec{\mathcal{S}}_n$, then $P = \sum T$ for some $T \subseteq \vec{\mathcal{S}}_n$.*

Proof:

Let $P \xrightarrow{a} P'$;

Then $\sum S \xrightarrow{a} P'' = P'$,

so $P_0 \xrightarrow{a} P''$ for some $P_0 \in S \subseteq \vec{\mathcal{S}}_n$;

But then by **Definition 5.3.1**, *$aP'' \in \vec{\mathcal{S}}_n$;*

Thus letting $T = \{aP'' \in \vec{\mathcal{S}}_n \mid \exists P' = P'' \text{ st } P \xrightarrow{a} P'\}$

we have $P = \sum T$. □

Corollary 5.3.4 *If $P = \sum S$ for some $S \subseteq \vec{\mathcal{S}}_n$, and $P \xrightarrow{a^j} P'$ for some $j > 0$, then there is some $R \in \vec{\mathcal{S}}_n$ such that $R = P'$.*

Proof:

Suppose $P \xrightarrow{a} P'' \xrightarrow{a^{j-1}} P'$;

Then by **Definition 5.3.1**, *$\sum S \xrightarrow{a} P_0 = P''$ for some $P_0 \in \vec{\mathcal{S}}_n$;*

Hence by **Definition 5.3.1**, *$P_0 \xrightarrow{a^{j-1}} R = P'$ for some $R \in \vec{\mathcal{S}}_n$.* □

Another technical property which this set satisfies in which we shall be interested is given by the following proposition.

Proposition 5.3.5 *Let $m > 2$, and $0 < r_1 < r_2 < \cdots < r_m$. If there is some $P \in \vec{\mathcal{S}}_n$ such that for some Q, $\mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m} + Q = P$, with $|P| \leq n$, then*

$$P = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m} + Q = \varphi_n.$$

Proof:

Straightforward check through all of the possibilities for $P \in \vec{\mathcal{S}}_n$ given by the alternate definition of $\vec{\mathcal{S}}_n$ of **Proposition 5.3.2.** □

We now again make a definition of a property of equations in which we shall be interested, similar to that in the previous section.

Definition 5.3.6 *For $U, V \subseteq \mathcal{P}_2^0$ being two sets of terms, let us define $\Theta_n^L(U,V)$ to be the proposition which states the following:*

$$\begin{aligned} P \in U \cup V \implies & P \equiv \tilde{P}, \quad \text{and} \quad P \not\equiv \mathbf{0},\ P' + P'', \\ & \text{and} \quad \textstyle\sum U = \sum V = \sum S \quad \text{for some } S \subseteq \vec{\mathcal{S}}_n, \\ & \text{and} \quad \exists P \in U \text{ st } P = \varphi_2 \parallel \varphi_n, \\ & \text{and} \quad \nexists Q \in V \text{ st } Q = \varphi_2 \parallel \varphi_n. \end{aligned}$$

Thus $\Theta_n^L(U,V)$ states (among other things) that the equation $\sum U = \sum V$ expresses a (valid) equality between terms in which the term $\varphi_2 \parallel \varphi_n$ is captured by a single summand on the left hand side of the equality, but not by any single summand on the right hand side.

Then let $\Theta_n(U,V) = \Theta_n^L(U,V) \vee \Theta_n^L(V,U)$.

Proposition 5.3.7 *Let $n > 1$ and $U, V \subseteq \mathcal{P}_2^0$ be such that $\Theta_n(U,V)$, and let $P \in U \cup V$ be the term satisfying $P = \varphi_2 \parallel \varphi_n$; Then $P \equiv P_2 \parallel P_n$, where $P_2 = \varphi_2$ and $P_n = \varphi_n$.*

Proof:

$\varphi_2 \parallel \varphi_n \xrightarrow{a} \varphi_n$ *and* $\varphi_2 \parallel \varphi_n \xrightarrow{a} a \parallel \varphi_n \neq \varphi_n$;
Hence $P \not\equiv aP'$ *as* $aP' \xrightarrow{a} P'$ *only.*
Thus $P \equiv P' \parallel P''$ *where* $P', P'' \neq \mathbf{0}$;

Since φ_2 *and* φ_n *are prime, we must have that* P' *and* P'' *are precisely* φ_2 *and* φ_n.

Hence $P \equiv P_2 \parallel P_n$ *where* $P_2 = \varphi_2$ *and* $P_n = \varphi_n$. □

Proposition 5.3.8 *Let* t *be an open term in* $\mathcal{P}_2$, *and let* σ *be a substitution such that* $t\sigma \equiv \widetilde{t\sigma}$, *and such that for some* $x \in fv(t)$,

$$\sigma_x = a\varphi_n + aa\varphi_n;$$

Then $t\sigma \neq \varphi_2 \parallel \varphi_n$.

Proof:

Let t, σ *and* x *be as above;*

t *is of the form*

$$t \equiv t_1 + t_2 + \cdots + t_m,$$

where each $t_i \not\equiv t' + t''$;

Let k *be such that* $x \in fv(t_k)$;

Thus $t_k \not\equiv \mathbf{0}$;

If $t_k \equiv bt'$, *then* $x \in fv(t')$, *so* $|t\sigma| > |t'\sigma| \geq |\sigma_x| = n + 2$;

But $|\varphi_2 \parallel \varphi_n| = n + 2$, *so* $t\sigma \neq \varphi_2 \parallel \varphi_n$;

If $t_k \equiv t' \parallel t''$, *then* $x \in fv(t')$ *or* $x \in fv(t'')$,

so $|t\sigma| = |t'\sigma| + |t''\sigma| \geq |\sigma_x| = n + 2$;

so again $t\sigma \neq \varphi_2 \parallel \varphi_n$;

Finally, if $t_k \equiv x$ *then* $t\sigma \xrightarrow{a} a\varphi_n$;

but $\varphi_2 \parallel \varphi_n \not\xrightarrow{a} a\varphi_n$, *so again* $t\sigma \neq \varphi_2 \parallel \varphi_n$. □

Proposition 5.3.9 *Let* $\mathcal{F}$ *be a finite saturated set of sound axioms, and let* n *be bigger than twice the number of operators in any axiom in* $\mathcal{F}$. *Then no axiom* $t = u$ *in* $\mathcal{F}$ *can be instantiated to a statement* $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *such that* $\Theta_n(U, V)$.

Proof:

Let n *be as above, and suppose* $t = u$ *is an axiom in* $\mathcal{F}$ *such that under substitution* σ, $t = u$ *instantiates to* $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *such that* $\Theta_n(U, V)$..

Without loss of generality, assume that $\Theta_n^L(U, V)$;

Clearly, $fv(t) = fv(u)$, *as* $t = u$ *is assumed to be a valid axiom.*

$t \equiv t_1 + t_2 + \cdots + t_k$ *and* $u \equiv u_1 + u_2 + \cdots + u_{k'}$ *for some* $k, k' > 0$

where each $t_i, u_i \not\equiv v + v'$;

$\Theta_n^L(U, V) \implies$ *for some* i, *either* $t_i\sigma \equiv P_2 \parallel P_n$ *or* $t_i\sigma \equiv P_2 \parallel P_n + Q$,

where $P_2 = \varphi_2$ *and* $P_n = \varphi_n$;

Consider the structure of t_i:

$t_i \equiv \mathbf{0} \implies t_i\sigma \equiv \mathbf{0}$ (contradiction);

$t_i \equiv x \implies \sigma_x \equiv t_i\sigma$ *and* $x \in fv(u_j)$ *for some* j

$\implies u_j \not\equiv \mathbf{0}, au', u' + u'', u' \parallel u''$

$\implies u_j \equiv x$ *and* $P_2 \parallel P_n \in V$

(contradicting $\Theta_n^L(U, V)$);

$t_i \equiv at' \implies t_i\sigma \equiv a(t'\sigma)$ (contradiction);

$t_i \equiv t' + t'' \implies$ (contradiction);

Thus $t_i \equiv t' \parallel t''$ *and* $t_i\sigma \equiv t'\sigma \parallel t''\sigma \equiv P_2 \parallel P_n$;

Hence $t_i \equiv t' \parallel t''$ *with* $t'\sigma \equiv P_2 = \varphi_2$ *and* $t''\sigma \equiv P_n = \varphi_n$;

Now $t'' \equiv v_1 + v_2 + \cdots v_l$ *where* $l < \frac{n}{2}$ *and each* $v_h \not\equiv v + v'$;

$t''\sigma \equiv v_1\sigma + v_2\sigma + \cdots v_l\sigma = \varphi_n = \mathcal{A}_1 + \mathcal{A}_2 + \cdots + \mathcal{A}_n$,

so some $v_h\sigma = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$ *for some* $m > 2$ *and*
$0 < r_1 < r_2 < \cdots < r_m$;

Thus clearly $v_h \not\equiv \mathbf{0}, av, v + v', v \parallel v'$, *so* $v_h \equiv x$ *for some variable* x *where* $\sigma_x = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$;

Clearly $x \notin fv(t')$, *as* $|t'\sigma| = 2 < r_m = |\sigma_x|$;

Let $\sigma' = \sigma\left\{a\varphi_n + aa\varphi_n/x\right\}$;

Then $t'\sigma' \equiv t'\sigma$, *and* $t\sigma' \xrightarrow{a} t'\sigma' \parallel \varphi_n = \varphi_2 \parallel \varphi_n$;

Therefore for some j, $u_j\sigma' \xrightarrow{a} \varphi_2 \parallel \varphi_n$;

Now $|u_j\sigma'| > n + 2 = |u\sigma|$, *so clearly* $x \in fv(u_j)$;

Consider the structure of u_j:

$u_j \equiv \mathbf{0} \implies x \notin fv(u_j)$ (contradiction);

$u_j \equiv x \implies u_j\sigma' \equiv a\varphi_n + aa\varphi_n \not\xrightarrow{a} \varphi_2 \parallel \varphi_n$
(contradiction);

$u_j \equiv au' \implies u_j\sigma' \equiv a(u'\sigma')$
$\implies u'\sigma' = \varphi_2 \parallel \varphi_n$ *and* $x \in fv(u')$
(contradiction) *(by* **Proposition 5.3.8***)*

$u_j \equiv u' + u'' \implies$ (contradiction);

Hence $u_j \equiv u' \parallel u''$ *with* $u''\sigma' \xrightarrow{a} p$ *st* $u'\sigma' \parallel p = \varphi_2 \parallel \varphi_n$;

If $x \in fv(u')$,

then $n + 2 = |u'\sigma'| + |p| \geq |\sigma'_x| + |p| \geq n + 2 + |p|$;

so $p = \mathbf{0}$ *and* $u'\sigma' = \varphi_2 \parallel \varphi_n$;

(contradicting **Proposition 5.3.8***)*

Hence $x \notin fv(u')$, *and so* $x \in fv(u'')$;

Now $u'\sigma \parallel p = \varphi_2 \parallel \varphi_n$, *and* $u'\sigma \neq \mathbf{0}$,

so $u'\sigma = \varphi_2$ *or* $u'\sigma = \varphi_n$ *or* $u'\sigma = \varphi_2 \parallel \varphi_n$;

But $|u'\sigma| = |u_j\sigma| - |u''\sigma| \leq |u\sigma| - |\sigma_x| < (n+2) - 2 = n$;

Therefore $u'\sigma = \varphi_2$;

Thus also $|u''\sigma| \leq n$;

Now, $x \in fv(u'') \implies u''\sigma \xrightarrow{a^j} \sigma_x + Q$ *for some* Q, $j \geq 0$;

Hence $u\sigma \xrightarrow{a^{j+1}} \sigma_x + Q$;

Thus by **Proposition 5.3.4**, $\exists P \in \vec{\mathcal{S}}_n$ *st* $\sigma_x + Q = P$;

But $\sigma_x = \mathcal{A}_{r_1} + \mathcal{A}_{r_2} + \cdots + \mathcal{A}_{r_m}$ *for some* $m > 2$

with $0 < r_1 < r_2 < \cdots < r_m$;

Hence by **Proposition 5.3.5**, $\sigma_x + Q = \varphi_n$;

Thus $u''\sigma \xrightarrow{a^j} \varphi_n$;

Now $n \geq |u''\sigma| \geq j + n$, *so* $j = 0$;

Therefore $u''\sigma = \varphi_n$;

But then $u'\sigma \parallel u''\sigma \equiv P_2 \parallel P_n \in V$ *for some* $P_2 = \varphi_2$ *and* $P_n = \varphi_n$

(contradicting $\Theta_n^L(U, V)$);

Therefore no axiom $t = u$ *in* $\mathcal{F}$ *can be instantiated to a statement* $p = q$ *where* $p \equiv \sum U$ *and* $q \equiv \sum V$ *such that* $\Theta_n(U, V)$. □

Proposition 5.3.10 *Suppose in a sound proof, we have an inference:*

$$\frac{p = r \quad r = q}{p = q}(trans)$$

where $p \equiv \sum U$, $q \equiv \sum V$, $r \equiv \sum W$,

and $R \in W \implies R \equiv \tilde{R}$, *and* $R \not\equiv \mathbf{0}, R' + R''$;

Then

$$\Theta_n(U,V) \implies \Theta_n(U,W) \vee \Theta_n(W,V).$$

Similarly for the (sub_+) *rule; corresponding to the inference:*

$$\frac{p = q \quad p' = q'}{p + p' = q + q'}(sub_+)$$

where $p \equiv \sum U$, $q \equiv \sum V$, $p' \equiv \sum U'$, *and* $q' \equiv \sum V'$, *we have the result that*

$$\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U,V) \vee \Theta_n(U',V').$$

Proof:

Consider the (trans) *rule case:*

Assume $\Theta_n^L(U,V)$; *We know immediately that*

$$P \in U \cup V \cup W \implies P \equiv \tilde{P} \text{ and } P \not\equiv \mathbf{0}, P' \parallel P'',$$

and (from $\Theta_n^L(U,V)$, *and the soundness of the proof in which the inference appears) that for some* $S \subseteq \vec{\mathcal{S}}_n$,

$$\sum U = \sum V = \sum W = \sum S;$$

Now if $\nexists R \in W$ *st* $R = a \parallel \varphi_n$, *then clearly* $\Theta_n^L(U,W)$;
And if $\exists R \in W$ *st* $R = a \parallel \varphi_n$, *then clearly* $\Theta_n^L(W,V)$;
Similarly, $\Theta_n^L(V,U) \implies \Theta_n^L(W,U) \vee \Theta_n^L(V,W)$;
Hence $\Theta_n(U,V) \implies \Theta_n(U,W) \vee \Theta_n(W,V)$.

The (sub_+) *rule case is similarly straightforward:*

Assume $\Theta_n^L(U \cup U', V \cup V')$; *Again we know immediately that*

$$P \in U \cup U' \cup V \cup V' \implies P \equiv \tilde{P} \text{ and } P \not\equiv \mathbf{0}, P' + P'',$$

and that for some $S \subseteq \vec{\mathcal{S}}_n$,

$$\sum(U \cup U') = \sum(V \cup V') = \sum S;$$

and

$$\exists P \in U \cup U' \text{ such that } P = a \parallel \varphi_n,$$

Suppose this $P \in U$*; then from* $\Theta_n^L(U \cup U', V \cup V')$*, and the soundness of the proof in which the inference appears, and from* **Proposition 5.3.3**, *we have for some* $S' \subseteq \vec{S}_n$,

$$\sum U = \sum V = \sum S'$$

so clearly $\Theta_n^L(U, V)$;

And similarly, if this $P \in U'$, *then* $\Theta_n^L(U', V')$.

Similarly, $\Theta_n^L(V \cup V', U \cup U') \implies \Theta_n^L(V, U) \vee \Theta_n^L(V', U')$;

Hence $\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U, V) \vee \Theta_n(U', V')$. □

5.3.2 Main Result

Here we state and prove our main theorem, the nonexistence of a finite axiomatisation of any "reasonable" equivalence.

Theorem 5.3.11 *Let* $\mathcal{F}$ *be a finite saturated set of sound (with respect to any fixed reasonable congruence) axioms, and let* n *be large enough (as allowed by* **Proposition 5.3.9***) so that no axiom in* $\mathcal{F}$ *can be instantiated to express any truth* $p = q$ *where*

$$p \equiv \sum U \text{ and } q \equiv \sum V \text{ such that } \Theta_n(U, V).$$

Then our system cannot prove the statement

$$\varphi_2 \parallel \varphi_n + \sum_{i=1}^{2} \sum_{j=1}^{n} \mathcal{A}_i \parallel \mathcal{A}_j = \sum_{i=1}^{2} \mathcal{A}_i \parallel \varphi_n + \sum_{j=1}^{n} \varphi_2 \parallel \mathcal{A}_j.$$

Hence no finite complete axiom system can exist for any reasonable congruence which is at least as strong as strong congruence.

Proof:

Suppose we have a (shortest) proof of the statement

$$\varphi_2 \parallel \varphi_n + \sum_{i=1}^{2}\sum_{j=1}^{n} \mathcal{A}_i \parallel \mathcal{A}_j = \sum_{i=1}^{2} \mathcal{A}_i \parallel \varphi_n + \sum_{j=1}^{n} \varphi_2 \parallel \mathcal{A}_j$$

which involves no terms containing $\mathbf{0}$ *as a summand or a factor. The proof takes the following form:*

$$\frac{\mathcal{D}_0}{p = q}(rule),$$

where $p \equiv \sum U_0$ *and* $q \equiv \sum V_0$ *for*

$$U_0 = \{\varphi_2, \varphi_n\} \cup \{\mathcal{A}_i \parallel \mathcal{A}_j \mid 1 \leq i \leq 2, 1 \leq j \leq n\}$$

and

$$V_0 = \{a \parallel \varphi_n, aa \parallel \varphi_n\} \cup \{\varphi_2 \parallel \mathcal{A}_j \mid 1 \leq j \leq n\}$$

so clearly $\Theta_n(U_0, V_0)$ *holds.*

Since this must be a finite *proof, somewhere in the proof tree is an inference*

$$\frac{\mathcal{D}}{\sum U = \sum V}(rule) \qquad \text{where} \quad \Theta_n(U, V),$$

such that the premise $\mathcal{D}$ *of the inference contains no equality*

$$\sum U' = \sum V' \quad \text{where} \quad \Theta_n(U', V');$$

By **Proposition 5.3.10**, $(rule)$ *can be neither of* $(trans)$ *nor* (sub_+);

Furthermore, by **Proposition 5.3.9**, *we know that* $(rule)$ *cannot be* $(t = u)$ *for any axiom* $t = u \in \mathcal{F}$;

Also clearly $(rule)$ *cannot be* $(symm)$, *as* $\Theta_n(U, V) \iff \Theta_n(V, U)$;

Finally, $(rule)$ *cannot be any of* $(refl)$, (sub_α), *or* $(sub_\parallel)$, *as this would contradict* $\Theta_n(U, V)$;

Hence we have shown that the original statement cannot be proven.

□

Chapter 6

Sequencing with the 0 Process

Up until this point, we have been interested solely in CCS-based process algebras. In this chapter, we shall consider a slightly different range of process algebras, those containing not CCS action prefixing, but rather sequential composition in the form of a binary operator $\cdot$ which takes two process terms and produces the new process which performs the actions of the first term, followed by those of the second. Such process algebras include for instance those based on **ACP**, the *Algebra of Communicating Processes* of Bergstra and Klop (*e.g.,* [BER84], [BER85]), as well as that of Boudol and Castellani's *partial order semantics* ([BOU86]).

The major proponents of equational studies of process algebras are certainly by far the Dutch researchers developing the **ACP**-based algebras. Much has been developed and published within this framework so a study of equational axiomatisations for process algebras would certainly be lacking if due attention were not awarded this family of languages.

Throughout the development of the **ACP** algebras, the basic components of a process were atomic actions, given by a set **Act** as in **CCS**. The major difference from **CCS**, apart from using sequencing rather than action prefixing, was the lack of a **0** process term in their algebras. For the basic algebra of terms, these were really the only variations from the **CCS**-type algebras and equivalences; by defining a form of sequencing within the **CCS** framework, and imposing (strong and weak) observational congruences on the resulting algebras, we would arrive at exactly the **ACP** notions of congruences, which initially are defined by axiom

systems without a previous operational model-based intuition built into the equational axioms. Thus we see presentations of **ACP**-type languages defined as the initial models of equational theories which in the **CCS** framework are precisely the complete axiomatisations of the operationally-based congruences.

Recently, a **0** process has been incorporated into the **ACP** algebras, in the form of an *empty process term* ε ([VRA86]). At this point, the **CCS** and **ACP** notions diverge greatly. The equivalence defined in [VRA86] for the **ACP** algebras with the ε term is no longer immediately recognisable as an observational congruence; the operational intuitions on which the **CCS** notion of equivalence is based is not explicitly present.

In this chapter, we quickly review the relevant portions of the **ACP** algebras for our investigation — namely, the Basic Process Algebra **BPA**, before and after the addition of the **0**-type process ε. From this brief exposé, we motivate a small change in the semantics of terms in order to bring the equivalence more in line with the usual **CCS**-based observational equivalences. We thus define a new semantic equivalence on terms, and examine questions regarding axiomatisations for our modified system. We then compare our system with $\mathbf{BPA}^{\varepsilon}$, first defining the $\mathbf{BPA}^{\varepsilon}$ congruence as an observational congruence, and then considering the relative ease which we have in adding merge and communication to our system, a state which as we shall see is not so true with $\mathbf{BPA}^{\varepsilon}$. Finally, we finish off this chapter with a modified proof of the previous chapter on the nonexistence of a finite axiomatisation for the **BPA** congruence when the full merge operator is added to the signature. To do this, we point out first that the unique decomposition result of **Chapter 4** holds in this algebra.

6.1 Introduction

In this section, we introduce the process algebra **BPA** (*Basic Process Algebra*), and the modified language $\mathbf{BPA}^{\varepsilon}$ involving the empty process ε introduced by [VRA86]. After the definitions of these algebras and the equivalences defined

for them are presented, we make some observations and arguments for modifying the equational system $\mathbf{BPA}^{\epsilon}$ from where it stands to something mimicking more closely the concepts and intuitions of **CCS**.

For a long time, researchers working on the **ACP** family of process algebras have dealt with languages not involving a **0**-like process. Their basic process algebra **BPA** is similar to a subset of **CCS**, with action prefixing replaced by atomic action processes and sequential composition, without the **0** process. They give a set of axioms, and define the equivalence within the process system as given by the initial model for these axioms. They present this system by means of an algebraic specification $\mathbf{BPA} = (\Sigma_{\mathbf{BPA}}, \mathbf{E}_{\mathbf{BPA}})$ in the style of [EHR85], as presented in **Figure 6–1**. Notice by convention that the sequential composition

$\Sigma_{\mathbf{BPA}}$

Sort	P	(processes)
Functions	$+ : P \times P \to P$	(choice)
	$\cdot : P \times P \to P$	(sequential execution)
Constants	$a \in \mathbf{Act}$	(atomic actions)

$\mathbf{E}_{\mathbf{BPA}}$

$$x + y = y + x$$
$$(x + y) + z = x + (y + z)$$
$$x + x = x$$
$$(x + y)z = xz + yz$$
$$(xy)z = x(yz)$$

Figure 6–1: Specification of $\mathbf{BPA} = (\Sigma_{\mathbf{BPA}}, \mathbf{E}_{\mathbf{BPA}})$

operator $\cdot$ will often be dropped from terms, thus being represented simply by the juxtaposition of terms. The congruence generated by these axioms corresponds precisely to strong observational congruence, as it would intuitively be defined in this context, and as we shall define it in the next section.

In [VRA86], this algebra is extended by the addition of a **0**-like process, the *empty process* ε. When the empty process ε is added to the signature, the modified system $\mathbf{BPA}^{\varepsilon} = (\Sigma_{\mathbf{BPA}^{\varepsilon}}, \mathbf{E}_{\mathbf{BPA}^{\varepsilon}})$ is defined as presented in **Figure 6–2.**

$\Sigma_{\mathbf{BPA}^{\varepsilon}}$

Sort	P	(processes)
Functions	$+ : P \times P \to P$	(choice)
	$\cdot : P \times P \to P$	(sequential execution)
Constants	ε	(empty process)
	$a \in \mathbf{Act}$	(atomic actions)

$\mathbf{E}_{\mathbf{BPA}^{\varepsilon}}$

$$x + y = y + x$$
$$(x + y) + z = x + (y + z)$$
$$(x + y)z = xz + yz$$
$$(xy)z = x(yz)$$
$$\varepsilon + \varepsilon = \varepsilon$$
$$\varepsilon x = x$$
$$x\varepsilon = x$$

Figure 6–2: Specification of $\mathbf{BPA}^{\varepsilon} = (\Sigma_{\mathbf{BPA}^{\varepsilon}}, \mathbf{E}_{\mathbf{BPA}^{\varepsilon}})$

In going from **BPA** to $\mathbf{BPA}^{\varepsilon}$, a few points immediately arise. Firstly, we appear to lose the idempotence axiom,

$$x + x = x.$$

However, in fact, this law is derivable in the system $\mathbf{E}_{\mathbf{BPA}^{\varepsilon}}$, using distributivity and the ε-laws. Secondly, if we expected the ε process to be intuitively analogous to the **CCS 0** process, we quickly discover that there is no **0**-absorptive law of the form

$$x + \varepsilon = x.$$

This law cannot in fact be proven, and in the system $\mathbf{BPA}^{\varepsilon}$ is not valid, for if it were, we could for example prove (using the distributivity of $\cdot$ over $+$ on the right) that

$$ab = (a+\varepsilon)b = ab + \varepsilon b = ab + b,$$

which we clearly would not desire. In the system $\mathbf{BPA}^{\varepsilon}$, the option of doing nothing (performing the ε process, allowing termination) is considered to be a valid choice. In **CCS**, such a process is liken to the process $\tau.\mathbf{0}$. Hence the approach taken by $\mathbf{BPA}^{\varepsilon}$ is motivated more by the work done on **CSP** ([HOA85]), where a **0**-like process exists representing *termination*, allowing an unobservable termination, or "tick", action $\surd$. In this case, one might expect that the process $a + \varepsilon$ might be allowed to terminate silently in the context

$$(a + \varepsilon) \between b,$$

allowing the term to evolve silently into the process term b. However if this were to be the case, we could argue that in the $\mathbf{BPA}^{\varepsilon}$ framework we would arrive at a nonassociative parallel composition operator $\between$. Indeed, in [VRA86] we find remarks giving evidence that an earlier attempt at introducing the empty process ε into the process algebras of **ACP** ran up against this very problem.

Thus the equivalence given by $\mathbf{BPA}^{\varepsilon}$ (or the empty process ε) neither resembles much that of **CSP** nor that of CCS's strong observational congruence (*i.e.*, the distributivity law fails here, whereas the absorptive law is valid). A new notion of *ε-bisimilarity* is introduced in [VRA86] to mimic the $\mathbf{BPA}^{\varepsilon}$ equivalence given by $\mathbf{E}_{\mathbf{BPA}^{\varepsilon}}$ using a modified notion of bisimulation. However, we are interested here in giving an axiom system for true observational congruence of their system, with the understanding that ε can perform no observable actions, thus restoring the operational intuition given by CCS. Indeed, we find that we need the operationally-based semantic understanding in order to proceed with the proofs of

our results on the unique decomposability and the nonexistence of finite axiomatisations for the original language of **BPA** with the full merge operator added to its signature, as shall be presented later in this chapter.

6.2 CCS With Sequencing

The first step towards our goal is to define our language and its operational semantics as a labelled transition system in the usual fashion so that we can define our congruence using the usual notion of bisimulation. This much is almost straightforward.

The terms in our language are exactly as given by the signature $\Sigma_{\mathbf{BPA}^\epsilon}$ in **Figure 6–2.** The process ϵ will represent the usual **0** process of **CCS.** Hence the only new notion here is the sequential composition operator.

The operational semantics for this languages will be given as usual by a transition system $\longrightarrow \subseteq \mathcal{P} \times \mathbf{Act} \times \mathcal{P}$ defined to be the least relation satisfying a certain collection of laws. In this case, we would intuitively like the transition system to satisfy the following laws:

$$\frac{}{a \xrightarrow{a} \epsilon}\ (a \in \mathbf{Act});$$

$$\frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'};\qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'};$$

$$\frac{p \xrightarrow{a} p'}{p \cdot q \xrightarrow{a} p' \cdot q};\qquad \frac{\forall b, p' : p \not\xrightarrow{b} p', \quad q \xrightarrow{a} q'}{p \cdot q \xrightarrow{a} q'}.$$

However, in light of the negative information in the premise of the last law, we cannot define our transition system to be the *least* relation satisfying these laws. For instance, if we added another law to the above, namely for some $a \in \mathbf{Act}$,

$$\frac{}{\epsilon \xrightarrow{a} \epsilon},$$

then the subprocess q in the process $\varepsilon\cdot q$ would never proceed, as we would have $\varepsilon\cdot q \xrightarrow{a} \varepsilon\cdot q$. A relation which satisfied this extended set of laws would of course satisfy our original set of laws, but would in fact be incomparable to the relation which we are defining. For instance, we would not want $\varepsilon\cdot b \xrightarrow{a} \varepsilon\cdot b$, but we would want $\varepsilon\cdot b \xrightarrow{b} \varepsilon$. Hence it would not make sense to require the existence of the least relation satisfying the original set of laws, as we have just exhibited two incomparable relations which each satisfy them, for neither of which is it the case that there exist a subset satisfying the laws. So we are forced to proceed in another way.

Our solution results from the fact that we can easily tell purely syntactically if a process is equivalent to the empty process ε. Thus we need simply to define syntactically when a term is equivalent to the ε process, and then to use this definition in the place of the negative information in the premise of the definition of our transition system above.

Proceeding in this way, we define the function *isempty* as follows:

$$\begin{aligned}
&\mathit{isempty}(\varepsilon) \;=\; \mathit{true};\\
&\mathit{isempty}(a) \;=\; \mathit{false} \qquad (a \in \mathbf{Act});\\
&\mathit{isempty}(p+q) \;=\; \mathit{isempty}(p\cdot q) \;=\; \mathit{isempty}(p) \wedge \mathit{isempty}(q).
\end{aligned}$$

Intuitively, we would like it to be the case that $p \sim \varepsilon$ *iff* $\mathit{isempty}(p)$, where $\sim$ is the strong observational equivalence which we are trying to define. Then our transition system can be validly defined as the least relation satisfying the following laws:

$$\frac{}{a \xrightarrow{a} \varepsilon}(a \in \mathbf{Act});$$

$$\frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'}; \qquad \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'};$$

$$\frac{p \xrightarrow{a} p'}{p\cdot q \xrightarrow{a} p'\cdot q}; \qquad \frac{\mathit{isempty}(p), \quad q \xrightarrow{a} q'}{p\cdot q \xrightarrow{a} q'}.$$

With the above transition system defined, our strong observational congruence $\sim$ is defined in the usual way via the notion of strong bisimulation using this

transition system. That is, $\sim$ is the largest relation such that whenever $P \sim Q$, we have that for all $a \in \mathbf{Act}$,

(i) $P \xrightarrow{a} P'$ implies $\exists Q' \sim P'$ such that $Q \xrightarrow{a} Q'$; and

(ii) $Q \xrightarrow{a} Q'$ implies $\exists P' \sim Q'$ such that $P \xrightarrow{a} P'$.

We can now check that our definition of *isempty* indeed satisfies our desired requirement.

Proposition 6.2.1 *$p \sim \varepsilon$ if and only if isempty(p).*

Proof:

Straightforward.

With the problems faced in defining the transition system above, it is worth checking that this definition does in fact yield a congruence relation. This is in fact not too difficult, as is outlined in the following proposition.

Proposition 6.2.2 *$\sim$ is a congruence.*

Proof:

That $\sim$ is an equivalence relation is straightforward, as it is defined via the notion of a bisimulation.

To check that it is substitutive, we need to confirm that the following inferences are valid:

$$\frac{p_0 \sim q_0, \quad p_1 \sim q_1}{p_0 + p_1 \ \sim \ q_0 + q_1}(sub_+), \qquad \frac{p_0 \sim q_0, \quad p_1 \sim q_1}{p_0 \cdot p_1 \ \sim \ q_0 \cdot q_1}(sub_\bullet).$$

The (sub_+) case is straightforward;

Assume that $p_0 \sim q_0$ and $p_1 \sim q_1$;

Suppose that $p_0 + p_1 \xrightarrow{a} p$;

Then either $p_0 \xrightarrow{a} p$ *or* $p_1 \xrightarrow{a} p$;

Hence for some $q \sim p$,

either $q_0 \xrightarrow{a} q$ *or* $q_1 \xrightarrow{a} q$;

Therefore $q_0 + q_1 \xrightarrow{a} q$;

Similarly, $q_0 + q_1 \xrightarrow{a} q$ *implies* $p_0 + p_1 \xrightarrow{a} p$ *for some* $p \sim q$;

Hence $p_0 + p_1 \sim q_0 + q_1$.

The $(sub_\bullet)$ *case is more involved, relying on a proof by induction on the depths of terms;*

Assume that $p_0 \sim q_0$ *and* $p_1 \sim q_1$, *and that for all* p'_0, p'_1, q'_0, q'_1 *such that* $p'_0 \sim q'_0$, $p'_1 \sim q'_1$, *and* $|p'_0| + |p'_1| < |p_0| + |p_1|$, *we have that* $p'_0 \cdot p'_1 \sim q'_0 \cdot q'_1$;

Suppose that $p_0 \cdot p_1 \xrightarrow{a} p$;

If $p_0 \xrightarrow{a} p'$ *such that* $p = p' \cdot p_1$,

then $q_0 \xrightarrow{a} q'$ *for some* $q' \sim p'$,

so $q_0 \cdot q_1 \xrightarrow{a} q' \cdot q_1 \sim p' \cdot p_1 = p$;

(by the inductive hypothesis)

If $p_0 \sim \varepsilon$ *and* $p_1 \xrightarrow{a} p$,

then $q_0 \sim \varepsilon$ *and* $q_1 \xrightarrow{a} q$ *for some* $q \sim p$,

so $q_0 \cdot q_1 \xrightarrow{a} q \sim p$;

Similarly, $q_0 \cdot q_1 \xrightarrow{a} q$ *implies* $p_0 \cdot p_1 \xrightarrow{a} p$ *for some* $p \sim q$;

Hence $p_0 \cdot p_1 \sim q_0 \cdot q_1$. □

Having succeeded in defining our observational congruence in the fashion which we desired, we can now proceed to its equational characterisation and properties.

6.2.1 Equational Axiomatisation

A complete axiomatisation for our congruence $\sim$ would naturally contain the axioms characterising the usual language of nondeterministic terms, that is:

$$(A_1)\ \ (x+y)+z = x+(y+z) \qquad (A_3)\ \ x+x = x$$
$$(A_2)\ \ x+y = y+x \qquad (A_4)\ \ x+\varepsilon = x$$

But we now need other axioms to deal with properties of sequencing (which never arose with action prefixing). In particular we need the three axioms

$$(S_1)\ \ (xy)z = x(yz) \qquad (S_2)\ \ x\cdot\varepsilon = x \qquad (S_3)\ \ \varepsilon\cdot x = x$$

as well as the axiom *schema*

$$(D_m)\ \ \Big(\sum_{i=1}^{m}\alpha_i x_i\Big)\cdot y = \sum_{i=1}^{m}\alpha_i x_i y \qquad (\textit{for } m>0,\ \alpha_i \in \mathbf{Act})$$

The motivation for introducing this axiom schema is that it mimics the distributivity law $(x+y)z = xz+yz$ of $\mathbf{E}_{\mathbf{BPA}^\varepsilon}$. This law holds as long as $x, y \neq \varepsilon$, and this is certainly true if x and y are both prefixed by atomic action terms. In this case though we must account for an arbitrary number of action-prefixed summands in the first term.

We shall refer to the above set of laws collectively by $\mathcal{T}$; that is, we let

$$\mathcal{T} = \big\{(A_1),(A_2),(A_3),(A_4),(S_1),(S_2),(S_3)\big\} \cup \big\{(D_m) \mid m>0\big\}.$$

That these laws are all valid is a straightforward thing to prove, and we shall take this fact to be granted. However, there are problems about the completeness of these axioms which we would now like to address, namely:

1. Are these laws complete for reasoning about closed terms?
2. Are these laws ω-complete, that is, complete for reasoning about open terms?
3. In view of the axiom schema (D_m) introduced, is the equivalence in this signature finitely axiomatisable for reasoning about closed terms, using some other axioms to replace the axiom schema above? If so, is it also finitely axiomatisable for open term reasoning?

The next section will answer the first question in the affirmative, giving a completeness proof for reasoning about closed terms. The following section will then answer the first part of the third question also in the affirmative, giving a finite axiomatisation for this system. After that, we shall investigate the axiomatisability of the extensional theory in this system, presenting some open statements which cannot be proven in the closed theories we develop.

6.2.2 Completeness for Closed Term Reasoning

The proof that the above laws $\mathcal{T}$ completely characterise the theory of closed terms over the language in question relies on a normal form for terms defined as usual by *derivation trees.* To get at the normal form, we shall again employ a denotation function which gives a set theoretic representation of the derivation tree corresponding to a given term.

A derivation tree again is simply a finite unordered tree whose arcs are labelled by elements of the action set **Act**, in which no two identically-labelled arcs lead out of the same node to two isomorphic subtrees. Such a model is well-known to be a complete characterisation of closed finite **CCS** terms, including the terms in our present framework. The domain of derivation trees can be represented formally as the least fixed point solution $\mathcal{D}$ to the set equation

$$\mathcal{D} = \mathcal{P}_{FIN}(\mathbf{Act} \times \mathcal{D})$$

where $\mathcal{P}_{FIN}(S)$ again represents the set of finite subsets of S.

Some of our proofs will depend on induction on the *depth* of the derivation trees of terms, which as usual in our set-theoretical formulation corresponds basically to the *rank* of a set. With this in mind, we define the rank of a derivation tree as follows.

Definition 6.2.3 *The* rank *of an element $P \in \mathcal{D}$ is defined inductively by*

$$rank(P) = \max\left(\{0\} \cup \{1 + rank(P') \mid \exists a \; st \; (a, P') \in P\}\right)$$

In particular, $rank(P) = 0$ *iff* $P = \emptyset$, *and for any* P, P' *where there is some* a *such that* $(a, P') \in P$, *we have that* $rank(P) > rank(P')$.

Our denotation defined for a term will simply be the set representation of the derivation tree representing the term. The definition of the denotation is straightforward, except in the case of sequential composition. Before giving the definition of the denotation on terms, we must state what the sequential composition of two derivation trees is. Intuitively, it is simply the first derivation tree with a copy of the second derivation tree tacked onto each of its leaves. More formally, for $P, Q \in \mathcal{D}$, we define $P \cdot Q$ inductively as follows:

$$\begin{aligned} \emptyset \cdot Q &= Q, \\ P \cdot Q &= \left\{(a, P' \cdot Q) \mid (a, P') \in P\right\} \quad (P \neq \emptyset). \end{aligned}$$

With this we can now define the denotation $[\![\cdot]\!] : \mathcal{P} \longrightarrow \mathcal{D}$ by cases on the structure of terms as follows:

$$\begin{aligned} [\![\varepsilon]\!] &= \emptyset; & [\![p + q]\!] &= [\![p]\!] \cup [\![q]\!]; \\ [\![a]\!] &= \{(a, \emptyset)\} \quad (a \in \mathbf{Act}); & [\![p \cdot q]\!] &= [\![p]\!] \cdot [\![q]\!]. \end{aligned}$$

An important property of this denotation, which is reflected in the derivation trees which the denotations represent, is given by the following proposition.

Proposition 6.2.4 $[\![p]\!] = \left\{(a, [\![p']\!]) \mid p \xrightarrow{a} p'\right\}$.

Proof:

By structural induction on p.

- $[\![\varepsilon]\!] = \emptyset = \left\{(a, [\![p']\!]) \mid \varepsilon \xrightarrow{a} p'\right\}$; *(as* $\varepsilon \not\xrightarrow{a} p'$ *for any* a, p'*)*
- $[\![b]\!] = \left\{(b, \emptyset)\right\} = \left\{(a, [\![p']\!]) \mid b \xrightarrow{a} p'\right\}$; *(as* $b \xrightarrow{b} \varepsilon$ *only)*
- $$\begin{aligned} [\![p + q]\!] &= [\![p]\!] \cup [\![q]\!] \\ &= \left\{(a, [\![p']\!]) \mid p \xrightarrow{a} p'\right\} \cup \left\{(a, [\![p']\!]) \mid q \xrightarrow{a} p'\right\} \\ &= \left\{(a, [\![p']\!]) \mid p + q \xrightarrow{a} p'\right\}; \end{aligned}$$
  (as $p + q \xrightarrow{a} p'$ *iff* $p \xrightarrow{a} p'$ *or* $q \xrightarrow{a} p'$*)*

- *If* $[\![p]\!] = \emptyset$, *then by the inductive hypothesis,*

  $p \not\xrightarrow{a} p'$ *for any* a, p', *so* $isempty(p)$;

  thus

$$\begin{aligned}[\![p\cdot q]\!] &= [\![p]\!]\cdot[\![q]\!] = \emptyset\cdot[\![q]\!] = [\![q]\!] \\ &= \{(a,[\![p']\!]) \mid q \xrightarrow{a} p'\} \\ &\qquad\qquad \textit{(by the inductive hypothesis)} \\ &= \{(a,[\![p']\!]) \mid p\cdot q \xrightarrow{a} p'\}; \\ &\qquad\qquad \textit{(by definition of } \xrightarrow{a}\textit{)}\end{aligned}$$

  If $[\![p]\!] \neq \emptyset$, *then by the inductive hypothesis,*

  $p \xrightarrow{a} p'$ *for some* a, p', *and hence* $\neg isempty(p)$;

  thus

$$\begin{aligned}[\![p\cdot q]\!] &= [\![p]\!]\cdot[\![q]\!] \\ &= \{(a,[\![p']\!]\cdot[\![q]\!]) \mid p \xrightarrow{a} p'\} \\ &\qquad\qquad \textit{(by the inductive hypothesis)} \\ &= \{(a,[\![p'\cdot q]\!]) \mid p \xrightarrow{a} p'\} \\ &= \{(a,[\![p']\!]) \mid p\cdot q \xrightarrow{a} p'\}. \\ &\qquad\qquad \textit{(by definition of } \xrightarrow{a}\textit{)}\end{aligned}$$

  □

A unique (modulo associativity and commutativity of +) normal form can be extracted from the denotation of a term in the following fashion:

$$nf(p) = \sigma([\![p]\!]),$$

where

$$\sigma(P) = \sum_{(a,P')\in P} a\cdot\sigma(P').$$

As usual, by convention, we let $\sigma(\emptyset) = \varepsilon$. That a term can be equated to its normal form relies on the following proposition.

Proposition 6.2.5 For $P, Q \in \mathcal{D}$, we have

$$\mathcal{T} \vdash \sigma(P)\cdot\sigma(Q) = \sigma(P\cdot Q).$$

Proof:

By induction on rank(P).

If rank(P) = 0, that is P = ∅, then the result is immediate, as

$$\sigma(\emptyset)\cdot\sigma(Q) = \varepsilon\cdot\sigma(Q) =_{\mathcal{T}} \sigma(Q) = \sigma(\emptyset\cdot Q).$$

For P ≠ ∅, we have that

$$\begin{aligned}
\sigma(P)\cdot\sigma(Q) &= \Big(\sum_{(a,P')\in P} a\cdot\sigma(P')\Big)\cdot\sigma(Q) \\
&=_{\mathcal{T}} \sum_{(a,P')\in P} a\cdot\Big(\sigma(P')\cdot\sigma(Q)\Big) \qquad \textit{(using } (D_m)\textit{)} \\
&=_{\mathcal{T}} \sum_{(a,P')\in P} a\cdot\Big(\sigma(P'\cdot Q)\Big) \\
&\qquad\qquad \textit{(by the inductive hypothesis)} \\
&= \sigma\Big(\{(a,P'\cdot Q) \mid (a,P')\in P\}\Big) \\
&= \sigma(P\cdot Q).
\end{aligned}$$

□

Proposition 6.2.6 $\mathcal{T} \vdash p = nf(p)$.

Proof:

By structural induction on p.

- $\varepsilon = \sigma(\emptyset) = \sigma([\![\varepsilon]\!]) = nf(\varepsilon)$;
- $a =_{\mathcal{T}} a\cdot\varepsilon = \sigma(\{(a,\emptyset)\}) = \sigma([\![a]\!]) = nf(a)$;
- $p =_{\mathcal{T}} nf(p) = \sigma([\![p]\!])$ *and* $q =_{\mathcal{T}} nf(q) = \sigma([\![q]\!])$

$$\begin{aligned}
\Longrightarrow p+q &=_{\mathcal{T}} \sigma([\![p]\!]) + \sigma([\![q]\!]) \\
&=_{\mathcal{T}} \sigma([\![p]\!] \cup [\![q]\!]) \\
&= \sigma([\![p+q]\!]) = nf(p+q);
\end{aligned}$$

- $p =_T nf(p) = \sigma([\![p]\!])$ *and* $q =_T nf(q) = \sigma([\![q]\!])$
  $\Longrightarrow p \cdot q =_T \sigma([\![p]\!]) \cdot \sigma([\![q]\!])$
  $=_T \sigma([\![p]\!] \cdot [\![q]\!])$ *(by* **Proposition 6.2.5***)*
  $= \sigma([\![p \cdot q]\!]) = nf(p \cdot q).$ □

Corollary 6.2.7 $[\![p]\!] = [\![q]\!] \Longrightarrow \mathcal{T} \vdash p = q.$

Proof:

$$p =_T nf(p) = \sigma([\![p]\!]) = \sigma([\![q]\!]) = nf(q) =_T q.$$ □

Finally, the proof of completeness of the axioms relies on the following proposition.

Proposition 6.2.8 $p \sim q \Longrightarrow [\![p]\!] = [\![q]\!].$

Proof:

We shall actually show by induction on rank$([\![p]\!])$ *that*

$$[\![p]\!] \neq [\![q]\!] \Longrightarrow p \not\sim q.$$

Thus suppose that $[\![p]\!] \neq [\![q]\!]$*, and that (without loss of generality)* $(a, P) \in [\![p]\!]$ *but* $(a, P) \notin [\![q]\!]$*;*

Then by **Proposition 6.2.4**, $\exists p'$ *st* $P = [\![p']\!]$ *and* $p \xrightarrow{a} p'$*;*

Suppose that $q \xrightarrow{a} q'$*;*

Then again by **Proposition 6.2.4**, $(a, [\![q']\!]) \in [\![q]\!]$*;*

Hence since $(a, P) \notin [\![q]\!]$, $[\![q']\!] \neq P = [\![p']\!]$*;*

Therefore by the inductive hypothesis, $p' \not\sim q'$*;*

Thus $\nexists q' \sim p'$ *st* $q \xrightarrow{a} q'$*, so* $p \not\sim q.$ □

Corollary 6.2.9 (Completeness) $p \sim q \Longrightarrow \mathcal{T} \vdash p = q.$

Proof:

Follows from **Proposition 6.2.8** *and* **Corollary 6.2.7.** □

6.2.3 A Finite Axiomatisation

In this section we shall present a finite axiomatisation for closed terms of our language. What we shall do is present two new rules which will together subsume the power of the axiom schema (D_m). That is, we shall show that given the other sum and sequencing laws, together with these two new laws, we can prove any instance of the schema (D_m).

The two laws which we need are as follows. Firstly we have an absorption law

$$(Abs) \quad (\alpha x + y)z = (\alpha x + y)z + \alpha xz,$$

which is very similar to the absorption law of **CCS**

$$(\alpha x + y) \parallel z = (\alpha x + y) \parallel z + \alpha(x \parallel z),$$

introduced in **Section 3.3**. Secondly we have a reduction law

$$(Red) \quad (x + y + z)\omega + y\omega + z\omega = (x + y)\omega + y\omega + z\omega.$$

Again these two new laws are easily seen to be valid. Notice here that if we let $x = \epsilon$, we get the derived reduction law

$$(Red') \quad (y + z)\omega + y\omega + z\omega = y\omega + z\omega,$$

which we shall use in the proof of the following proposition.

Proposition 6.2.10 *Let*

$$\mathcal{F} = \{(A_1), (A_2), (A_3), (A_4), (S_1), (S_2), (S_3), (Abs), (Red)\};$$

then for any $m > 0$, $\mathcal{F} \vdash (D_m)$.

Proof:

By induction on m.

- *For $m = 1$, the result follows from (S_1) (associativity of $\cdot$);*

- *For $m = 2$, we have*

$$(\alpha_1 x_1 + \alpha_2 x_2)y =_{\mathcal{F}} (\alpha_1 x_1 + \alpha_2 x_2)y + \alpha_1 x_1 y + \alpha_2 x_2 y$$

(using (Abs) twice)

$$=_{\mathcal{F}} \alpha_1 x_1 y + \alpha_2 x_2 y;$$

(using the derived law (Red'))

- *For $m > 2$, we have*

$$\Big(\sum_{i=1}^{m} \alpha_i x_i\Big)y$$

$$=_{\mathcal{F}} \Big(\sum_{i=1}^{m-2} \alpha_i x_i + \alpha_{m-1} x_{m-1} + \alpha_m x_m\Big)y + \alpha_{m-1} x_{m-1} y + \alpha_m x_m y$$

(using (Abs) twice)

$$=_{\mathcal{F}} \Big(\sum_{i=1}^{m-1} \alpha_i x_i\Big)y + \alpha_{m-1} x_{m-1} y + \alpha_m x_m y$$

(using (Red))

$$=_{\mathcal{F}} \sum_{i=1}^{m-1} \alpha_i x_i y + \alpha_{m-1} x_{m-1} y + \alpha_m x_m y$$

(by the inductive hypothesis)

$$=_{\mathcal{F}} \sum_{i=1}^{m} \alpha_i x_i y. \qquad \square$$

Thus since by the previous section, $\mathcal{F} \cup \{(D_m) \mid m > 0\}$ is a complete theory for reasoning about closed terms over this language, we have shown that the finite theory $\mathcal{F}$ is itself complete for closed term reasoning, as every instance of (D_m) is derivable from $\mathcal{F}$.

In fact, as pointed out by Frits Vaandrager, we could have replaced the schema (D_m) by just one law, namely

$$(ax+by+z)w = axw + (by+z)w.$$

6.2.4 Reasoning About Open Terms

The question which we wish to address now is that of the axiomatisability of the theory of open terms over this language. In particular we wish to know if we can ω-completely axiomatise the theory, and if so if we can do it with a finite number of axioms.

Our original system for closed term reasoning, involving the axiom schema (D_m) was clearly not complete for reasoning about open terms, as it could not prove the validity of the two laws we introduced in the previous section, namely

$$(Abs)\quad (\alpha x + y)z = (\alpha x + y)z + \alpha xz;$$
$$(Red)\quad (x + y + z)\omega + y\omega + z\omega = (x + y)\omega + y\omega + z\omega.$$

The form of the law (Red) makes analysis of the system for open terms complicated, as it is not immediately clear where to look for a normal form for this system in light of this law.

If we consider the forms of the (Abs) law and the derived law,

$$(Red')\quad (x + y)\omega + x\omega + y\omega = x\omega + y\omega$$

we see that we can introduce some notion of *saturation* to attempt to get a grasp on a normal form. That is, a normal form can be assumed to contain every instance of αxz as a summand whenever it contains $(\alpha x + y)z$ as a summand (from the (Abs) law), and to contain every instance of $(x + y)\omega$ as a summand whenever it contains $x\omega$ and $y\omega$ as summands (from the (Red') law). But saturation using the (Red) law does not fall out so easily.

However, let us consider the form of the derived law (Red') more closely. It very much resembles the classic *testing equivalence* law from [DEN84],

$$\alpha(x + y) + \alpha x + \alpha y = \alpha x + \alpha y.$$

If we consider the analogous law corresponding to the second of the two classic testing equivalence laws, namely

$$\alpha(x + y + z) + \alpha x = \alpha(x + y + z) + \alpha(x + y) + \alpha x,$$

then we get the following new law

$$(Red'')\quad (x + y + z)\omega + x\omega = (x + y + z)\omega + (x + y)\omega + x\omega,$$

which again is easily confirmed to be a valid law. However, it appears not to be derivable from the previous laws including (*Red*); but along with the law (*Red'*), it is indeed as strong as (*Red*), as shown by the following proposition.

Proposition 6.2.11 $\{(A_1),(A_2),(A_3),(A_4),(Red'),(Red'')\} \vdash (Red)$.

Proof:

$$\begin{aligned}
&(x+y+z)\omega + y\omega + z\omega \\
&\quad = (x+y+z)\omega + (x+y)\omega + y\omega + z\omega && \textit{(by (Red''))} \\
&\quad = \big((x+y)+z\big)\omega + (x+y)\omega + z\omega + y\omega \\
&\quad = (x+y)\omega + z\omega + y\omega && \textit{(by (Red'))} \\
&\quad = (x+y)\omega + y\omega + z\omega. && \square
\end{aligned}$$

Thus we can (and do) replace the original law (*Red*) by the two laws (*Red'*) and (*Red''*) to get a more powerful system. Furthermore, the newest law (*Red''*) also fits nicely into our scheme of saturating terms to derive a normal form; that is, now saturation will include the condition that a term contain every instance of $(x+y)\omega$ as a summand whenever it contains $x\omega$ and $(x+y+z)\omega$ as summands.

The question now is whether these laws completely characterise the open theory. If this were so, then our saturation technique could perhaps be used to define a canonical form for expressions to facilitate a completeness proof. Alas though, we can show that we still do not have a complete system. For instance, we cannot prove the following valid open statements:

$$(xax'+y)z = (xax'+y)z + xax'z$$

$$\big((ax+x')y+z\big)\omega = \big((ax+x')y+z\big)\omega + (ax+x')y\omega$$

In the general case, we want our absorption law to extend to cover the *pseudo*-inference law:

$$\frac{\mathrm{Sort}(t) \neq \emptyset \quad (i.e.,\ t\{\bar{\varepsilon}/\bar{x}\} \not\simeq \varepsilon)}{(t+u)v = (t+u)v + tv}$$

To accomplish this, it would appear that we need an infinite set of new axioms. However, we can accomplish the same result using the three conditional axioms:

$$(C_1)\quad \frac{(t+u)\omega = (t+u)\omega + t\omega}{(t+t'+u)\omega = (t+t'+u)\omega + (t+t')\omega};$$

$$(C_2)\quad \frac{(t+u)\omega = (t+u)\omega + t\omega}{(st+u)\omega = (st+u)\omega + st\omega};$$

$$(C_3)\quad \frac{(t+u)\omega = (t+u)\omega + t\omega}{(ts+u)\omega = (ts+u)\omega + ts\omega}.$$

Proposition 6.2.12 *(C_1), (C_2) and (C_3) above are sound conditional laws.*

Proof:

We shall only deal with case (C_3), as the other two cases are identical.

We just need show that for closed terms p, p', q, r and P,

if $(p+p')q \sim (p+p')q + pq$, and $prq \xrightarrow{a} P$,
then $(pr+p')q \xrightarrow{a} Q$ for some $Q \sim P$.

If $\neg$isempty(pr),

then $pr \xrightarrow{a} p_0$ st $P = p_0 q$;
hence $(pr+p')q \xrightarrow{a} p_0 q = P$.

If isempty(pr),

then $q \xrightarrow{a} P$ and isempty(p);
thus $pq \xrightarrow{a} P$, so $(p+p')q \xrightarrow{a} Q$ for some $Q \sim P$;
hence either $p' \xrightarrow{a} p_0$ st $Q = p_0 q$,
whence $(pr+p')q \xrightarrow{a} Q$;
or isempty(p') and $q \xrightarrow{a} Q$,
whence isempty$(pr+p')$ and so $(pr+p')q \xrightarrow{a} Q$. □

These three conditional laws are in fact enough to give us our desired result, as shown by the following proposition.

Proposition 6.2.13 $\{(A_2),(C_1),(C_2),(C_3)\} \vdash \dfrac{\mathrm{Sort}(t) \neq \emptyset}{(t+u)v = (t+u)v + tv}$

Proof:

By structural induction on t.

- $t \equiv \varepsilon$ *or* $t \equiv x \implies \mathrm{Sort}(t) = \emptyset$;

- $t \equiv t_1 + t_2$ *and* $\mathrm{Sort}(t) \neq \emptyset \implies \mathrm{Sort}(t_1) \neq \emptyset$ *or* $\mathrm{Sort}(t_2) \neq \emptyset$;

  $\mathrm{Sort}(t_1) \neq \emptyset \implies (t_1+u)v = (t_1+u)v + t_1v$
  (by the inductive hypothesis)
  $\implies (t+u)v = (t+u)v + tv$; *(using* C_1*)*

  $\mathrm{Sort}(t_2) \neq \emptyset \implies (t+u)v = (t+u)v + tv$.
  (Similarly, using A_2*)*

- $t \equiv t_1t_2$ *and* $\mathrm{Sort}(t) \neq \emptyset \implies \mathrm{Sort}(t_1) \neq \emptyset$ *or* $\mathrm{Sort}(t_2) \neq \emptyset$;

  $\mathrm{Sort}(t_1) \neq \emptyset \implies (t_1+u)v = (t_1+u)v + t_1v$
  (by the inductive hypothesis)
  $\implies (t+u)v = (t+u)v + tv$; *(using* C_3*)*

  $\mathrm{Sort}(t_2) \neq \emptyset \implies (t+u)v = (t+u)v + tv$;
  (Similarly, using C_2*)*

□

Thus we find ourselves using unconditional axioms to cover all of the possible cases implied by the above conditional rule. Having done this, we are still left with a nontrivial task of finding a canonical form for terms to show that we now have an ω-complete axiomatisation, which we leave here as a still open problem.

6.3 Comparison with $\mathbf{BPA}^{\varepsilon}$

We have now redefined the semantics of the process algebra $\mathbf{BPA}^{\varepsilon}$ to define the equivalence between terms as an observational congruence. Unlike the original theory for $\mathbf{BPA}^{\varepsilon}$ presented as a simple equational theory in **Figure 6–2**, the distributivity law $(x + y)z = xz + yz$ is not valid in our semantic model. However, we have discovered a simple finite set $\mathcal{F}$ of equational laws characterising the new congruence, so the original theory for $\mathbf{BPA}^{\varepsilon}$ does not much benefit over this new framework in ease of equational presentation.

In [VRA86], a graph model is presented which characterises the $\mathbf{BPA}^{\varepsilon}$ congruence which uses a modified notion of bisimulation, called *ε-bisimulation.* What we shall show in this section is a method of characterising the $\mathbf{BPA}^{\varepsilon}$ congruence as a true observational congruence, by defining a natural labelled transition system on terms of the algebra which when used as the basis for a bisimulation equivalence will not exactly be the congruence of $\mathbf{BPA}^{\varepsilon}$, but will contain this congruence as the largest congruence within it. However the characterisation is difficult to prove, as the defined bisimulation equivalence is not a congruence, and several alternate characterisations of the congruence will need to be invoked in the proof.

We shall then consider including the merge operators, first without and then with communication, into the signature, and compare how our new semantic model and the original $\mathbf{BPA}^{\varepsilon}$ model of [VRA86] fare in this experiment.

6.3.1 The $\mathbf{BPA}^{\varepsilon}$ Equivalence as an Observational Congruence

In this section, we shall present a bisimulation characterisation of the $\mathbf{BPA}^{\varepsilon}$ equivalence described in **Section 6.1**. That is, we shall present a transition system defined on our process algebra such that the largest congruence contained in the bisimulation equivalence defined by the transition system will be precisely the congruence generated by the axioms of $\mathbf{E}_{\mathbf{BPA}^{\varepsilon}}$.

Similar to the definition *isempty* of the previous section, we can define a "*hasempty*" predicate on terms specifying when a term contains the empty process ε (or anything equivalent to it) as a summand, as follow.

$$\begin{aligned}
&hasempty(\varepsilon) = true;\\
&hasempty(a) = false \quad (a \in \mathbf{Act});\\
&hasempty(p+q) = hasempty(p) \vee hasempty(q);\\
&hasempty(p{\cdot}q) = hasempty(p) \wedge hasempty(q).
\end{aligned}$$

With this predicate, we can define a transition system $\longrightarrow$ as the least relation satisfying the following laws.

$$\begin{aligned}
&a \xrightarrow{a} \varepsilon \quad (a \in \mathbf{Act}), \quad \text{and}\\
&P \xrightarrow{a} P' \Longrightarrow P+Q \xrightarrow{a} P',\ Q+P \xrightarrow{a} P',\ P{\cdot}Q \xrightarrow{a} P'{\cdot}Q, \text{ and}\\
&\qquad\qquad hasempty(Q) \Longrightarrow Q{\cdot}P \xrightarrow{a} P'.
\end{aligned}$$

From here we could define an equivalence $\approx$ in the usual fashion via bisimulations by letting $\approx$ be the largest binary relation such that whenever $P \approx Q$ we have that for all $a \in \mathbf{Act}$,

(i) $P \xrightarrow{a} P'$ implies $\exists Q' \approx P'$ such that $Q \xrightarrow{a} Q'$; and

(ii) $Q \xrightarrow{a} Q'$ implies $\exists P' \approx Q'$ such that $P \xrightarrow{a} P'$.

However this equivalence relation is not a congruence. This is because we would have for instance $a+\varepsilon \approx a$ but $(a+\varepsilon)a \not\approx aa$. Hence we define the relation which we are interested in to be the largest congruence $\approx^c$ contained in $\approx$.

This congruence relation, as we shall show, is precisely that of the $\mathbf{BPA}^\varepsilon$ system. Unfortunately, it is not a straightforward task to prove that this congruence coincides with that defined by the axioms of $\mathbf{E}_{\mathbf{BPA}^\varepsilon}$. To do this, we use yet another characterisation, which we now describe.

Let $\cong$ be the largest binary relation such that whenever $P \cong Q$, we have that for all $a \in \mathbf{Act}$,

(i) $P \xrightarrow{a} P'$ implies $\exists Q' \cong P'$ such that $Q \xrightarrow{a} Q'$;

(ii) $Q \xrightarrow{a} Q'$ implies $\exists P' \cong Q'$ such that $P \xrightarrow{a} P'$; and

(iii) $hasempty(P) = hasempty(Q)$.

We can easily show that this relation, defined by a modified bisimulation, is a congruence relation. Also, it is clearly contained in the equivalence relation $\approx$ (as it satisfies the definition of $\approx$, being defined by a more restrictive definition). We shall continue from here to show that $\cong$ is in fact precisely the congruence defined by $\mathbf{E}_{\mathbf{BPA}^\epsilon}$. Upon doing that, we shall show that $\approx^c$ is contained in $\cong$. From this it shall follow (since $\approx^c$ is the largest congruence contained in $\approx$) that $\approx^c$ coincides precisely with $\cong$, and so also with the congruence generated by $\mathbf{E}_{\mathbf{BPA}^\epsilon}$.

That the equations of $\mathbf{E}_{\mathbf{BPA}^\epsilon}$ are valid $\cong$-equivalences is a straightforward matter to verify. The proof that these laws completely characterise the equivalence $\cong$ relies on a normal form for terms defined by *flagged derivation trees*, where the flags are used to specify when a process has the empty process as a summand. To get at the normal form, we shall once again employ a denotation function which gives a set theoretic representation of the flagged derivation tree corresponding to a given term.

A flagged derivation tree is simply a finite unordered tree whose arcs are labelled by elements of the action set **Act**, and whose nodes are labelled by the set $\{true, false\}$, in which no two identically-labelled arcs lead out of the same node to two isomorphic subtrees. As usual, the arcs eminating from the root of a flagged derivation tree represent the possible actions which the corresponding process can perform, namely, those which label the arcs. The flag associated with the derivation tree specifies whether or not the empty process is a summand in the process term represented by the tree.

The domain of flagged derivation trees can be represented formally as the least fixed point solution $\mathcal{D}$ to the set equation

$$\mathcal{D} = \mathcal{P}_{FIN}(\mathbf{Act} \times \mathcal{D}) \times \{true, false\}$$

where $\mathcal{P}_{FIN}(S)$ again represents the set of finite subsets of S. For an element $D = (T, tt) \in \mathcal{D}$, let

$$tree(D) = T, \quad \text{and} \quad flag(D) = tt.$$

Again, some of our proofs will depend on induction on the *depth* of the flagged derivation trees of terms, which in our set-theoretical framework corresponds basically to the *rank* of a set. With this in mind, we make the following definition.

Definition 6.3.1 *The* rank *of an element $D \in \mathcal{D}$ is defined inductively by*

$$rank(D) = \max\Big(\{0\} \cup \{1 + rank(D') \mid \exists a \text{ st } (a, D') \in tree(D)\}\Big)$$

Thus in particular, $rank(D) = 0$ iff $D = (\emptyset, tt)$, and for any D, D' where there exists some a such that $(a, D') \in tree(D)$, we have that $rank(D) > rank(D')$.

Our denotation defined for a term will simply be the set representation of the flagged derivation tree representing the term. Before giving the definition we must state what the sequential composition of two flagged derivation trees is. Intuitively, it is simply the first flagged derivation tree with a copy of the second flagged derivation tree tacked onto each node which is labelled by *true*, with the flag of that node reset to equal the flag of the second tree. More formally, for $P, Q \in \mathcal{D}$, we define $P{\cdot}Q$ inductively as follows:

$$\begin{aligned} tree(P{\cdot}Q) &= \big\{(a, P'{\cdot}Q) \mid (a, P') \in tree(P)\big\} \\ &\quad \cup \big\{(a, Q') \mid (a, Q') \in tree(Q) \wedge flag(P) = true\big\}; \\ flag(P{\cdot}Q) &= flag(P) \wedge flag(Q). \end{aligned}$$

With this we can now define the denotation $[\![\cdot]\!] : \mathcal{P} \longrightarrow \mathcal{D}$ by cases on the structure of terms as follows:

$$\begin{aligned} [\![\epsilon]\!] &= (\emptyset, true); \\ [\![a]\!] &= \big(\{(a, (\emptyset, true))\}, false\big), \quad (a \in \mathbf{Act}); \\ [\![p + q]\!] &= \big(tree([\![p]\!]) \cup tree([\![q]\!]),\ flag([\![p]\!]) \vee flag([\![q]\!])\big); \\ [\![p{\cdot}q]\!] &= [\![p]\!]{\cdot}[\![q]\!]. \end{aligned}$$

An important property of this denotation, which is reflected in the flagged derivation trees which the denotations represent, is given by the following proposition.

Proposition 6.3.2 $[\![p]\!] = \big(\{(a, [\![p']\!]) \mid p \xrightarrow{a} p'\}, hasempty(p)\big)$.

Proof:

By structural induction on p;

- $[\![\varepsilon]\!] = (\emptyset, true) = \big(\{(a, [\![p']\!]) \mid \varepsilon \xrightarrow{a} p'\}, hasempty(\varepsilon)\big)$;
  (as $\varepsilon \not\xrightarrow{a} p'$ for any a, p' and $hasempty(\varepsilon) = true$)

- $[\![b]\!] = \big(\{(b, (\emptyset, true))\}, false\big)$
  $= \big(\{(a, [\![p']\!]) \mid b \xrightarrow{a} p'\}, hasempty(b)\big)$;
  (as $b \xrightarrow{b} \varepsilon$ only, and $hasempty(b) = false$)

- $[\![p + q]\!] = \big(tree([\![p]\!]) \cup tree([\![q]\!]), flag([\![p]\!]) \vee flag([\![q]\!])\big)$
  $= \big(\{(a, [\![p']\!]) \mid p \xrightarrow{a} p'\} \cup \{(a, [\![q']\!]) \mid q \xrightarrow{a} q'\},$
  $hasempty(p) \vee hasempty(q)\big)$
  (by the inductive hypothesis)
  $= \big(\{(a, [\![r']\!]) \mid p + q \xrightarrow{a} r'\}, hasempty(p + q)\big)$;

- $[\![p \cdot q]\!] = [\![p]\!] \cdot [\![q]\!]$
  $= \big(\{(a, P' \cdot [\![q]\!]) \mid (a, P') \in tree([\![p]\!])\}$
  $\cup \{(a, Q') \mid (a, Q') \in tree([\![q]\!])$
  $\wedge flag([\![p]\!]) = true\},$
  $flag([\![p]\!]) \wedge flag([\![q]\!])\big)$
  $= \big(\{(a, [\![p']\!] \cdot [\![q]\!]) \mid p \xrightarrow{a} p'\}$
  $\cup \{(a, [\![q']\!]) \mid q \xrightarrow{a} q' \wedge hasempty(p) = true\},$
  $hasempty(p) \wedge hasempty(q)\big)$
  (by the inductive hypothesis)
  $= \big(\{(a, [\![p' \cdot q]\!]) \mid p \xrightarrow{a} p'\}$
  $\cup \{(a, [\![q']\!]) \mid q \xrightarrow{a} q' \wedge hasempty(p) = true\},$
  $hasempty(p \cdot q)\big)$
  $= \big(\{(a, [\![r]\!]) \mid p \cdot q \xrightarrow{a} r\}, hasempty(p \cdot q)\big)$.
  (by definition of $\xrightarrow{a}$)

□

A unique (modulo associativity and commutativity of + and associativity of · normal form can be extracted from the denotation of a term in the following fashion:

$$nf(p) = \sigma([\![p]\!])$$

where

$$\sigma(D) = \sum\Big(\{a{\cdot}\sigma(D') \mid (a, D') \in tree(D)\} \cup \{\varepsilon \mid flag(D) = true\}\Big).$$

That a term can be equated to its normal form relies on the following proposition.

Proposition 6.3.3 *For $P, Q \in \mathcal{D}$, we have*

$$\sigma(P){\cdot}\sigma(Q) = \sigma(P{\cdot}Q).$$

Proof:

By induction on rank(P).

$$\sigma(P){\cdot}\sigma(Q) = \Big(\sum\Big(\big\{a{\cdot}\sigma(P') \mid (a, P') \in tree(P)\big\} \cup \big\{\varepsilon \mid flag(P) = true\big\}\Big)\Big) \cdot \sigma(Q)$$

$$= \sum\Big(\big\{a{\cdot}(\sigma(P'){\cdot}\sigma(Q)) \mid (a, P') \in tree(P)\big\} \cup \big\{\sigma(Q) \mid flag(P) = true\big\}\Big)$$

(using distributivity)

$$= \sum\Big(\big\{a{\cdot}\sigma(P'{\cdot}Q) \mid (a, P') \in tree(P)\big\}$$

$$\cup \left\{a{\cdot}\sigma(Q') \;\middle|\; flag(P) = true \;\wedge\; (a, Q') \in tree(Q)\right\}$$

$$\cup \left\{\varepsilon \;\middle|\; flag(P) = flag(Q) = true\right\}\Big)$$

(by the inductive hypothesis)

$$= \Sigma\Big(\left\{a{\cdot}\sigma(R) \;\middle|\; (a, R) \in tree(P{\cdot}Q)\right\}$$

$$\cup \left\{\varepsilon \;\middle|\; flag(P{\cdot}Q) = true\right\}\Big)$$

$$= \sigma(P{\cdot}Q). \qquad \square$$

Proposition 6.3.4 $p = nf(p)$.

Proof:

By structural induction on p.

- $\varepsilon = \sigma((\emptyset, true)) = \sigma([\![\varepsilon]\!]) = nf(\varepsilon)$;
- $a = a{\cdot}\varepsilon = \sigma\Big(\big(\{(a, (\emptyset, true))\}, false\big)\Big)$

  $= \sigma([\![a]\!]) = nf(a)$;
- $p = nf(p) = \sigma([\![p]\!])$ *and* $q = nf(q) = \sigma([\![q]\!])$

  $\Longrightarrow p + q = \sigma([\![p]\!]) + \sigma([\![q]\!])$

  $= \sigma(tree([\![p]\!]) \cup tree([\![q]\!]),$

  $flag([\![p]\!]) \vee flag([\![q]\!]))$

  $= \sigma([\![p + q]\!]) = nf(p + q)$;
- $p = nf(p) = \sigma([\![p]\!])$ *and* $q = nf(q) = \sigma([\![q]\!])$

  $\Longrightarrow p{\cdot}q = \sigma([\![p]\!]){\cdot}\sigma([\![q]\!])$

  $= \sigma([\![p]\!]{\cdot}[\![q]\!])$

  (by **Proposition 6.3.3***)*

  $= \sigma([\![p{\cdot}q]\!]) = nf(p{\cdot}q)$. $\square$

Corollary 6.3.5 $[\![p]\!] = [\![q]\!] \implies p = q.$

Proof:

$$p = nf(p) = \sigma([\![p]\!]) = \sigma([\![q]\!]) = nf(q) = q. \qquad \square$$

Finally, the proof of completeness of the axioms relies on the following proposition.

Proposition 6.3.6 $p \cong q \implies [\![p]\!] = [\![q]\!].$

Proof:

We shall actually show by induction on rank$([\![p]\!])$ *that*

$$[\![p]\!] \neq [\![q]\!] \quad \textit{implies} \quad p \ncong q.$$

Thus suppose that $[\![p]\!] \neq [\![q]\!]$.

If $\mathit{flag}([\![p]\!]) \neq \mathit{flag}([\![q]\!])$, *then by* **Proposition 6.3.2** *we would have that* $\mathit{hasempty}(p) \neq \mathit{hasempty}(q)$, *so clearly* $p \ncong q$.

Hence we just need show that if $\mathit{tree}([\![p]\!]) \neq \mathit{tree}([\![q]\!])$ *then* $p \ncong q$.

Suppose then that $\mathit{tree}([\![p]\!]) \neq \mathit{tree}([\![q]\!])$, *and that (without loss of generality)* $(a, P) \in \mathit{tree}([\![p]\!])$ *but* $(a, P) \notin \mathit{tree}([\![q]\!])$;

Then by **Proposition 6.3.2**, $\exists p'$ *st* $P = [\![p']\!]$ *and* $p \xrightarrow{a} p'$;

Suppose that $q \xrightarrow{a} q'$;

> *Then again by* **Proposition 6.3.2**, $(a, [\![q']\!]) \in \mathit{tree}([\![q]\!])$;
> *Hence since* $(a, P) \notin \mathit{tree}([\![q]\!])$, $[\![q']\!] \neq P = [\![p']\!]$;
> *Therefore by the inductive hypothesis,* $p' \ncong q'$;

Thus $\nexists q' \cong p'$ *such that* $q \xrightarrow{a} q'$, *so* $p \ncong q$. $\square$

Corollary 6.3.7 (Completeness) $p \cong q \implies p = q.$

Proof:

Follows from **Proposition 6.3.6** *and* **Corollary 6.3.5.** □

It now remains to show that $\approx^c \subseteq \cong$. In order to do this, we shall show that $\approx^c$ satisfies the definition of $\cong$, and as such, since $\cong$ is defined to be the largest relation satisfying its definition, our result will follow.

Firstly it is straightforward to show that $\approx^c$ satisfies the final clause in the definition of $\cong$; namely, that $P \approx^c Q \implies hasempty(P) = hasempty(Q)$. This is true as

$hasempty(p) = false$ implies $\forall a, r \; st \; p \cdot b \xrightarrow{a} r, \exists p' \; st \; r = p' \cdot b$, but

$hasempty(p) = true$ implies $p \cdot b \xrightarrow{b} \varepsilon \neq p' \cdot b$.

Hence it remains to show that

$$p \approx^c q \text{ and } p \xrightarrow{a} p' \text{ implies } \exists q' \approx^c p' \text{ such that } q \xrightarrow{a} q'$$

and vice versa. This follows from the following final characterisation of $\approx^c$.

Proposition 6.3.8 *$p \approx q$ implies $\forall r : r \cdot p \approx r \cdot q$.*

Hence $\forall r : p \cdot r \approx q \cdot r \implies p \approx^c q$.

Proof:

By induction on depth(r). Let $p \approx q$. $r \cdot p \xrightarrow{a} p'$ iff either of the following:

(i) $r \xrightarrow{a} r' \wedge p' = r' \cdot p$ *or*

(ii) $hasempty(r) \wedge p \xrightarrow{a} p'$;

For (i), $r \xrightarrow{a} r' \wedge p' = r' \cdot p \implies r \cdot q \xrightarrow{a} r' \cdot q \approx r' \cdot p = p'$;

(by the inductive hypothesis)

For (ii), $hasempty(r) \wedge p \xrightarrow{a} p' \implies \exists q' \approx p' \; st \; q \xrightarrow{a} q'$

$\implies r \cdot q \xrightarrow{a} q' \approx p'$. □

Proposition 6.3.9 $p \approx^c q \wedge p \xrightarrow{a} p'$ *implies* $\exists q' \approx^c p'$ *such that* $q \xrightarrow{a} q'$.

Proof:

Let $p \approx^c q$ *and* $p \xrightarrow{a} p'$.

Furthermore, let r *be any arbitrary process.*

By **Proposition 6.3.8**, *we need only show that*

$$\exists q' \text{ st } q \xrightarrow{a} q' \wedge p' \cdot r \approx q' \cdot r.$$

But we know that $p \cdot r \approx q \cdot r$ *and* $p \cdot r \xrightarrow{a} p' \cdot r$;

(by **Proposition 6.3.8***)*

Hence $q \cdot r \xrightarrow{a} r' \approx p' \cdot r$;

Therefore $\exists q'$ *st* $q \xrightarrow{a} q'$ *and* $q' \cdot r \approx p' \cdot r$. □

Thus we have managed to characterise the $\mathbf{BPA}^{\varepsilon}$ congruence as an observational congruence. However, the approach ran upon difficult points which were not be met by the more natural congruence defined in the previous section. Furthermore, as we shall see in the next section, the original congruence does not fit as well into the theory as our new observational congruence when the parallel combinator is added to the signature of the language.

6.3.2 Adding Merge and Communication

In this section, we increase our language by adding the full merge $\|$ and left merge $\lfloor\!\lfloor$ operators, as is done in [VRA86]. The full merge operator is in fact the interesting new concept, whereas the left merge operator is added as usual simply to facilitate the easy (finite) axiomatisation of the resulting system.

The terms in our new language are thus given by the extended signature $\Sigma = \mathbf{Act} \cup \{\varepsilon, +, \cdot, \|, \lfloor\!\lfloor\}$, with the labelled transition system operational behaviour of the operators defined as usual. When we define the new bisimulation equivalence based on this extended transition system, we get the theory which is well

known to be characterised by the theory $\mathcal{F}$ given for the sequential language above, along with the new merge laws

$$(M_1)\ \ x \parallel y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x \qquad (M_4)\ \ (x+y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$$
$$(M_2)\ \ \varepsilon \lfloor\!\lfloor x = \varepsilon \qquad (M_5)\ \ (x \lfloor\!\lfloor y) \lfloor\!\lfloor z = x \lfloor\!\lfloor (y \parallel z)$$
$$(M_3)\ \ \alpha x \lfloor\!\lfloor y = \alpha(x \parallel y)$$

That these laws are valid is again straightforward to prove. That they along with the previous laws suffice to characterise the closed theory follows from the conservativity of the extension (the true statements in the original signature will be the same as before), and the fact that using the merge laws, you can express any term as a term in the original signature.

Hence in the case of $\sim$, we are done. However, the congruence defined in [VRA86], which admits the distributivity law, simplifying the axiomatisation of the sequential language, suffers with the addition of the merge operators. For instance, if we added the above set of merge laws as is, due to the axiom M_2 we would be able to derive

$$(a \parallel b)c = a(bc+c) + b(ac+c),$$

which is clearly undesirable. Thus we cannot reduce $\varepsilon \lfloor\!\lfloor x$ to ε. The situation is remedied in [VRA86] by the introduction into the signature of another constant δ, a symbol used in the full **ACP** language to represent *deadlock*. Along with this new constant, we have the following axioms:

$$(D_1)\ \ \delta + x = x, \qquad (D_2)\ \ \delta x = \delta.$$

With this, we can bypass the problem with the occurrences of ε in expanding out merged terms by letting $\varepsilon \lfloor\!\lfloor x = \delta$. However, if we were to use this law universally, we would run into an equally disasterous situation, as then we would be able to derive

$$(a \parallel b)c = ab\delta + ba\delta.$$

Thus care had to be taken in [VRA86]. There we find that the single law (M_2) above is replaced by the following list of laws:

$$\begin{array}{ll} (M_{21}) \quad \varepsilon \lfloor\!\lfloor \varepsilon = \varepsilon; & (M_{23}) \quad \varepsilon \lfloor\!\lfloor \alpha x = \delta; \\ (M_{22}) \quad \varepsilon \lfloor\!\lfloor \delta = \delta; & (M_{24}) \quad \varepsilon \lfloor\!\lfloor (x+y) = \varepsilon \lfloor\!\lfloor x + \varepsilon \lfloor\!\lfloor y; \\ & (M_{25}) \quad \delta \lfloor\!\lfloor x = \delta. \end{array}$$

Thus we can see that whereas in our new semantic model for this process algebra with sequencing we had no problems in introducing the new operators for merging processes, the approach in [VRA86] had to take care in how these new operators are dealt with in its model.

From here we can add communication in the usual **ACP** fashion. First we define a *communication function* $\gamma : \mathbf{Act} \times \mathbf{Act} \longrightarrow \mathbf{Act}$ satisfying $\gamma(a,b) = \gamma(b,a)$ and $\gamma(\gamma(a,b),c) = \gamma(a,\gamma(b,c))$. Then we add a communication operator $\ddagger$ to our signature, and add to our labelled transition system operational semantics the following rules:

$$\frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q'}{p \parallel q \xrightarrow{\gamma(a,b)} p' \parallel q'} \qquad \frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q'}{p \ddagger q \xrightarrow{\gamma(a,b)} p' \parallel q'}$$

Again it is straightforward for us to axiomatise this extended system; we simply replace (M_1) above by

$$(M_1') \quad x \parallel y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x + x \ddagger y$$

and add the usual communication laws

$$\begin{array}{ll} (C_1) \quad \alpha \ddagger \beta = \gamma(\alpha,\beta) & (C_4) \quad \varepsilon \ddagger x = \varepsilon \\ (C_2) \quad x \ddagger y = y \ddagger x & (C_5) \quad \alpha x \ddagger y = (\alpha \ddagger y) \lfloor\!\lfloor x \\ (C_3) \quad (x \ddagger y) \ddagger z = x \ddagger (y \ddagger z) & (C_6) \quad (x+y) \ddagger z = x \ddagger z + y \ddagger z \end{array}$$

Again, the approach taken in [VRA86] had to take care in how it handled its equivalence. Its axiomatisation again required the use of the deadlock constant δ, and the replacement of law (C_4) above with the following laws:

$$(C_4') \quad \varepsilon \ddagger x = \delta, \qquad (C_4'') \quad \delta \ddagger x = \delta.$$

In conclusion, though the approach to adding the empty process to **BPA** taken in [VRA86] allowing the right distributivity of $\cdot$ over $+$ eases the axiomatisation of the sequential language, problems arise with the addition of the merge operators. The study must resort to going outside of the signature to introduce a constant δ representing deadlock to be able to axiomatise the system. In our case, we retained the well-respected observational equivalence, and tackled the problem of replacing the distributivity law with valid laws which would completely characterise the equivalence; in the end we indeed succeeded in finding two simple axioms which would do just that. From there we had no problem with adding the merge operators and axiomatising the resulting equivalence within the signature.

6.4 The Non-Finite-Axiomatisability of $\mathbf{BPA}_{\parallel}$

In this section, we consider the theory $\mathbf{BPA}_{\parallel}$ of the algebra **BPA** extended with the full merge operator $\parallel$ but not the left merge operator $\lfloor\!\lfloor$. The congruence defined for this language by the **ACP** community and as presented above as an observational congruence coincide in the absence of the empty process ε. This theory is given by the equational theory

$$\begin{aligned} \mathcal{T} \;=\; & \{(A_1),(A_2),(A_3),(S_1),(D)\} \\ & \cup \{(Exp_{m,M,n,N}) \mid 0 \leq m \leq M,\ 0 \leq n \leq N,\ M,N > 0\} \end{aligned}$$

as specified as follows.

$$\begin{array}{ll} (A_1) \quad (x+y)+z = x+(y+z) & (S_1) \quad (xy)z = x(yz) \\ (A_2) \quad x+y = y+x & (D) \quad (x+y)z = xz+yz \\ (A_3) \quad x+x = x & \end{array}$$

$(Exp_{m,M,n,N})$

$$For\ t = \sum_{i=1}^{m} \alpha_i x_i + \sum_{i=m+1}^{M} \alpha_i$$

$$\text{and} \quad u = \sum_{j=1}^{n} \beta_j y_j + \sum_{j=n+1}^{N} \beta_j, \quad (M, N > 0),$$

$$t \parallel u = \sum_{i=1}^{m} \alpha_i \big(x_i \parallel u\big) + \sum_{i=m+1}^{M} \alpha_i u + \sum_{j=1}^{n} \beta_j \big(t \parallel y_j\big) + \sum_{j=n+1}^{N} \beta_j t$$

We shall show in this section that this infinite equational axiomatisation cannot be replaced by any finite axiomatisation. Our proof method will be similar to that of **Section 5.2**, except we shall have to use a more subtle argument. In particular, the proof in **Section 5.2** relied on the fact that in the algebra in question (that using action prefixing), we could not express the process term

$$a \parallel (a + aa + \cdots + a^n)$$

as a sum of fewer than $n+1$ terms without one of those terms itself being equivalent to the whole term. In this case, this result is no longer valid, as

$$a \parallel (a + aa + \cdots + a^n)$$

$$= a(a + aa + \cdots + a^n) + (a + aa + \cdots + a^n)a.$$

Thus we must be careful in how closely we mimic the proof of **Section 5.2.**

6.4.1 Preliminary Results

Before we tackle our problem, we must address a few formalities. Firstly, we are going to view $\mathbf{BPA}_{\parallel}$ as a CCS-like algebra, and work with our defined transition system over it, as well as exploit the properties of bisimulations. To do this we must augment the language with the empty process ε, to allow the transitions $a \xrightarrow{a} \varepsilon$. Then our equivalence over this augmented language will be defined as in **Section 6.2.** This equivalence, when restricted to process terms not involving the ε term yields precisely the $\mathbf{BPA}_{\parallel}$-congruence $\mathcal{T}$ from above.

Our proofs are often going to use induction on the *depths* $|\cdot|$ of terms as defined as follows.

Definition 6.4.1

$$|a| = 1; \qquad |p \cdot q| = |p| + |q|;$$
$$|p + q| = \max\big(|p|, |q|\big); \qquad |p \parallel q| = |p| + |q|.$$

We also extend the definition of *depth* to include $depth(\epsilon) = 0$. Some important properties of depth which we shall exploit in our inductive proofs are given by the following proposition.

Proposition 6.4.2 *For all* $p, q \in \mathbf{BPA}_{\parallel}$,

(*i*) $|p| > 0$;
(*ii*) $|p \cdot q| > |p|, |q|$;
(*iii*) $|p \parallel q| > |p|, |q|$.
(*iv*) $|p| = 1$ *iff* $p = a$ *for some* $a \in \mathbf{Act}$;
(*v*) $p \xrightarrow{a} p'$ *implies* $|p| > |p'|$;

Proof:

Straightforward. □

Next, we can define a couple special semantic classes of *prime* terms which shall be useful in our analysis. As before, they will give us a handle on syntactically classifying terms.

Definition 6.4.3 *A term* $p \in \mathbf{BPA}_{\parallel}$ *is* <u>seq-prime</u> *iff it cannot be expressed as* $p = q \cdot r$ *for any* $q, r \in \mathbf{BPA}_{\parallel}$. *A term* $p \in \mathbf{BPA}_{\parallel}$ *is* <u>prime</u> *iff it cannot be expressed as* $p = q \parallel r$ *for any* $q, r \in \mathbf{BPA}_{\parallel}$.

Simple tests for primality (seq-primality) are given by the following proposition.

Proposition 6.4.4

(*i*) *If* $p \xrightarrow{a} \epsilon$, *then* p *is both prime and seq-prime.*

(*ii*) *If* $p \xrightarrow{a} p'$ *and* $p \xrightarrow{b} p''$, *where* p' *and* p'' *are distinct seq-primes, then* p *itself must be a seq-prime.*

(*iii*) *If* $p \xrightarrow{a} p'$ *and* $p \xrightarrow{b} p''$, *where* p' *and* p'' *are distinct primes such that* $p \neq p' \parallel p''$, *then* p *itself must be a prime.*

(*iv*) *If* $p \xrightarrow{a} p'$, $p \xrightarrow{b} p''$ *and* $p \xrightarrow{\sigma} p'''$, *where* p', p'' *and* p''' *are distinct primes, then* p *itself must be a prime.*

Proof:

Straightforward. □

The useful result about these prime (seq-prime) terms is given by the following proposition regarding the decomposition of terms.

Proposition 6.4.5 (Unique Factorisation Theorems) *Any term* $p \in \mathbf{BPA}_{\parallel}$ *can be expressed uniquely as a sequential composition of seq-primes, and uniquely as a parallel composition of primes.*

Proof:

As in **Section 4.1** *or* **Section 4.2.** □

Generally, we shall work with processes defined using only a single atomic action symbol $a \in \mathbf{Act}$. As we explained above, this is the only assumption we shall make on the action set **Act**, that such an action exists, and this assumption is valid and necessary. Two important sequences of terms which we shall make extensive use of are given in the following definition.

Definition 6.4.6

$$\varphi_n = a + aa + \cdots + a^n;$$

$$\Phi_n = a\varphi_1 + a\varphi_2 + \cdots + a\varphi_n.$$

Example primes and seq-primes which will be useful to us are given by the following propositions.

Proposition 6.4.7 *For* $n > 1$,

(i) φ_n *is seq-prime;* *(iv)* $a \parallel \varphi_n$ *is seq-prime*

(ii) Φ_n *is seq-prime;* *(v)* $a \parallel \Phi_n$ *is seq-prime;*

(iii) Φ_n *is prime;* *(vi)* $a \parallel \Phi_n + Q$ *is seq-prime* $\forall Q$.

Proof:

Straightforward, using **Proposition 6.4.4.** □

Proposition 6.4.8 *For* $h > 1$ *and* $0 < r_1 < r_2 < \cdots < r_h$,

$$a\varphi_{r_1} + a\varphi_{r_2} + \cdots + a\varphi_{r_h}$$

is prime and seq-prime.

Proof:

Straightforward, using **Proposition 6.4.4** *and* **Proposition 6.4.7.**

□

The utility of these propositions will become evident in the proofs to follow. By saying that a term is prime (seq-prime), we are restricting the possible syntactic form of the term.

6.4.2 Technical Lemmata

In this section we state and prove the technical lemmata which we need to derive our main result in the final section. Firstly however, we define a proposition on pairs of sets of terms which will designate a property of equations which we want to analyse in our proof system.

Definition 6.4.9 *For* $n > 1$ *and* $U, V \subseteq \mathbf{BPA}_{\parallel}$ *being two sets of terms, let* $\Theta_n^L(U,V)$ *be the proposition which states the following:*

$$P \in U \cup V \implies P \not\equiv P' + P'',$$

$$\text{and } \sum U = \sum V = a \parallel \Phi_n,$$

$$\text{and } \exists P \in U \text{ st } P = a \parallel \Phi_n,$$

$$\text{and } \nexists Q \in V \text{ st } Q = a \parallel \Phi_n.$$

Thus $\Theta_n^L(U,V)$ states that the equation $\sum U = \sum V$ expresses a (valid) equality between terms equal to $a \parallel \Phi_n$ in which the term $a \parallel \Phi_n$ is already captured by a single summand on the left hand side of the equality, but not by any single summand on the right hand side of the equality.

Then let $\Theta_n(U,V) = \Theta_n^L(U,V) \vee \Theta_n^L(V,U)$.

Proposition 6.4.10 *Let $n > 1$ and $U, V \subseteq \mathbf{BPA}_\delta$ be such that $\Theta_n(U,V)$, and let $P \in U \cup V$ be the term satisfying $P = a \parallel \Phi_n$; Then $P \equiv A \parallel P_n$, where $A = a$ and $P_n = \Phi_n$.*

Proof:

$P \not\equiv P' \cdot P''$, as $P = a \parallel \Phi_n$ is seq-prime. *(by* **Proposition 6.4.7**(v)*)*

Also, $P \not\equiv a, P' + P''$.

Hence $P \equiv P' \parallel P''$.

Now $a \parallel \Phi_n \xrightarrow{a} \Phi_n$, which is prime; *(by* **Proposition 6.4.7**(ii)*)*

Without loss of generality, assume that $P' \xrightarrow{a} Q$ where $Q \parallel P'' = \Phi_n$;

Then $Q = \varepsilon$, $P'' = \Phi_n$, and $P = P' \parallel \Phi_n$;

Hence $P' = a$;

(by **Proposition 6.4.2**(iv)*, as $|P'| = |a \parallel \Phi_n| - |P''| = 1$)*

Thus $P \equiv A \parallel P_n$, where $A = a$ and $P_n = \Phi_n$. □

Proposition 6.4.11 *Let $\mathcal{F}$ be a finite set of (valid) equational axioms, and let n be bigger than twice the number of operators in any axiom in the set $\mathcal{F}$. Then no axiom $t = u$ in $\mathcal{F}$ can be instantiated to a statement $\sum U = \sum V$ where $\Theta_n(U,V)$.*

Proof:

Let n be as above, and suppose $t = u$ is an axiom in $\mathcal{F}$ such that under substitution σ, $t = u$ instantiates to $\sum U = \sum V$ where $\Theta_n(U, V)$.

Without loss of generality, assume that $\Theta_n^L(U, V)$;

Clearly, $fv(t) = fv(u)$, as $t = u$ is assumed to be a valid axiom.

$t \equiv t_1 + t_2 + \cdots + t_k$ *and* $u \equiv u_1 + u_2 + \cdots + u_{k'}$ *for some* $k, k' > 0$,

where each $t_i, u_i \not\equiv v + v'$;

$\Theta_n^L(U, V) \implies$ *for some* i, *either* $t_i\sigma \equiv A \parallel P_n$ *or* $t_i\sigma \equiv A \parallel P_n + Q$,

where $A = a$ *and* $P_n = \Phi_n$;

(by **Proposition 6.4.10***)*

Thus $t_i\sigma$ is seq-prime; *(by* **Proposition 6.4.7***)*

Consider the structure of t_i:

$t_i \equiv a \implies t_i\sigma \equiv a$ *(contradiction);*

$t_i \equiv t' \cdot t'' \implies t_i\sigma \equiv (t'\sigma)(t''\sigma)$

(contradiction, as t_i is seq-prime)

$t_i \equiv t' + t'' \implies$ *(contradiction);*

$t_i \equiv x \implies \sigma_x \equiv t_i\sigma$ *and* $x \in fv(u_j)$ *for some* j

$\implies u_j \not\equiv a$ *(as $x \notin fv(a)$)*

and $u_j \not\equiv u' + u''$ *(by assumption on u_j's)*

and $u_j \not\equiv u' \cdot u''$, $u' \parallel u''$

(as otherwise $n + 2 = |u\sigma| \leq |u_j\sigma| < |\sigma_x| = n + 2$*)*

$\implies u_j \equiv x$ *and* $A \parallel P_n \in V$

(contradicting $\Theta_n^L(U, V)$)

Thus $t_i \equiv t' \parallel t''$ *and* $t_i\sigma \equiv t'\sigma \parallel t''\sigma = a \parallel \Phi_n$;

Hence $t_i \equiv t' \parallel t''$ *with* $t'\sigma \equiv A = a$ *and* $t''\sigma \equiv P_n = \Phi_n$;

(by **Proposition 6.4.7**(iii) *and* **Proposition 6.4.5***)*

Now $t'' \equiv v_1 + v_2 + \cdots v_l$ *where* $l < \frac{n}{2}$ *and each* $v_h \not\equiv v + v'$;

$t''\sigma \equiv v_1\sigma + v_2\sigma + \cdots v_l\sigma = \Phi_n = a\varphi_1 + a\varphi_2 + \cdots + a\varphi_n$,

so some $v_h\sigma = a\varphi_{r_1} + a\varphi_{r_2} + \cdots + a\varphi_{r_a}$ *for some* $a > 2$, *and*

some $r_1, r_2, \cdots, r_a$ *st* $0 < r_1 < r_2 < \cdots < r_a$;

Thus clearly $v_h \not\equiv a, v + v', vv', v \parallel v'$,

(by primality/seq-primality of $v_h\sigma$, *from* **Proposition 6.4.8***)*

so $v_h \equiv x$ *for some variable* x

with $\sigma_x = a\varphi_{r_1} + a\varphi_{r_2} + \cdots + a\varphi_{r_a}$;

Clearly $x \notin fv(t')$, *as* $|t'\sigma| = |a| = 1 < r_a = |\sigma_x|$;

Let $\sigma' = \sigma\left\{{}^{a\Phi_n}/_x\right\}$;

Then $t'\sigma' \equiv t'\sigma$, *and* $t\sigma' \xrightarrow{a} t'\sigma' \parallel \Phi_n = a \parallel \Phi_n$;

Therefore for some j, $u_j\sigma' \xrightarrow{a} a \parallel \Phi_n$;

Now $|u_j\sigma'| > n + 2 = |u\sigma|$, *so clearly* $x \in fv(u_j)$;

Consider the structure of u_j:

$u_j \equiv x \implies u_j\sigma' \equiv a\Phi_n \not\xrightarrow{a} a \parallel \varphi_n$ *(contradiction);*

$u_j \equiv a \implies x \notin fv(u_j)$ *(contradiction);*

$u_j \equiv u' \cdot u'' \implies u_j\sigma \equiv (u'\sigma)(u''\sigma) \xrightarrow{a} a \parallel \Phi_n$,

which is prime;

$\implies u'\sigma' \xrightarrow{a} \varepsilon$ *and* $u''\sigma' = a \parallel \Phi_n$

with $x \in fv(u'')$;

(as $|u''\sigma| < |u''\sigma'|$*)*

$u'' \equiv \omega_1 + \omega_2 + \cdots + \omega_l$ *for some* l

with each $\omega_h \not\equiv \omega + \omega'$,

and $x \in fv(\omega_m)$ *for some* m;

Consider the structure of the ω_m *with* $x \in fv(\omega_m)$:

$\omega_m \equiv a \implies x \notin fv(\omega)$ *(contradiction);*

$\omega_m \equiv \omega + \omega' \implies$ *(contradiction);*

$\omega_m \equiv \omega \cdot \omega'$ *or* $\omega_m \equiv \omega \parallel \omega'$

$\implies x \in fv(\omega)$ *or* $x \in fv(\omega')$

$\implies n+2 = |u''\sigma'| \geq |\omega_m \sigma'|$

$= |\omega\sigma'| + |\omega'\sigma'|$

$> |\sigma'_x| = n+2$

(contradiction);

Thus $x \in fv(\omega_m) \implies \omega_m \equiv x$ *and* $\omega_m \sigma' = a\Phi_n$;

Now $u''\sigma' = a \parallel \Phi_n \xrightarrow{a} a \parallel \varphi_n$,

so $\omega_m \sigma' \xrightarrow{a} a \parallel \varphi_n$ *for some* m;

Clearly $\sigma'_x \not\xrightarrow{a} a \parallel \varphi_n$;

Hence $\omega_m \sigma' \xrightarrow{a} a \parallel \varphi_n$ *for some* m

such that $x \notin fv(\omega_m)$;

But then $\omega_m \sigma \xrightarrow{a} a \parallel \varphi_n$;

Thus $n+2 \leq |\omega_m \sigma| \leq |u''\sigma|$

$< |u_j \sigma| \leq |u\sigma| = n+2$

(contradiction)

$u_j \equiv u' + u'' \implies$ *(contradiction);*

Hence $u_j \equiv u' \parallel u''$ *with* $x \in fv(u')$;

Now since $u''\sigma \xrightarrow{a} p$ *for some* p, *we have* $u\sigma \xrightarrow{a} u'\sigma \parallel p$;

Thus $u'\sigma \parallel p = \Phi_n$ *or* $u'\sigma \parallel p = a \parallel \varphi_k$ *for some* $k : 1 \leq k \leq n$;

If $u'\sigma \parallel p = \Phi_n$, *then* $p = \varepsilon$ *and* $u'\sigma = \Phi_n$, *and so also* $u''\sigma = a$,

(since Φ_n *is prime, by* **Proposition 6.4.7**(*iii*)*)*

so $u_j\sigma \equiv u'\sigma \parallel u''\sigma = a \parallel \Phi_n$ *(contradicting* $\Theta^L_n(U,V)$*);*

Hence $u'\sigma \parallel p = a \parallel \varphi_k$ *for some* $k : 1 \leq k \leq n$;

Since $x \in fv(u')$ *and* $\sigma_x = a\varphi_{r_1} + a\varphi_{r_2} + \cdots + a\varphi_{r_a}$,

we have that for some $l > 0$ *and for some* $q \in \mathcal{P}$,

$u'\sigma \xrightarrow{a^l} \varphi_{r_s} q$ *for each* $s : 1 \leq s \leq a$;

Now for $l' > 1$, $u'\sigma \parallel p = a \parallel \varphi_k \xrightarrow{a^{l'}} p'$ *implies* $p' = a^s$

for some $s : 0 \leq s < k$;

But $u'\sigma \parallel p \xrightarrow{a^{l+l'}} \varphi_{r_a} q$, *where* $p \xrightarrow{a^{l'}} \varepsilon$,

and $\varphi_{r_a} q \neq a^s = aa \cdots a$, *(By* **Proposition 6.4.5***)*

so $l + l' < 2$, *which means* $l' = 0$ *and thus* $p = \varepsilon$;

Hence $u'\sigma = a \parallel \varphi_k$;

But then $u'\sigma = a \parallel \varphi_k \xrightarrow{a^l} \varphi_{r_s} q$ *implies* $r_s = 1$ *or* $r_s = k$;

(contradicting $a > 2$ *where* $u'\sigma \xrightarrow{a^l} \varphi_{r_s}$

for each $s : 1 \leq s \leq a$, *and* $0 < r_1 < r_2 < \cdots < r_a$*)*

Therefore no axiom $t = u \in \mathcal{F}$ *can be instantiated to a statement* $\sum U = \sum V$ *where* $\Theta_n(U, V)$. □

Proposition 6.4.12 *Suppose in a sound proof, we have an inference:*

$$\frac{\sum U = \sum W \quad \sum W = \sum V}{\sum U = \sum V}(trans)$$

where $R \in W \implies R \not\equiv R' + R''$; *Then*

$$\Theta_n(U, V) \implies \Theta_n(U, W) \vee \Theta_n(W, V).$$

Similarly for the (sub_+) *rule; corresponding to the inference:*

$$\frac{\sum U = \sum V \quad \sum U' = \sum V'}{\sum U \cup U' = \sum V \cup V'}(sub_+),$$

we have the result that

$$\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U, V) \vee \Theta_n(U', V').$$

Proof:

Consider the (trans) *rule case:*

Assume $\Theta_n^L(U,V)$;

We know immediately that

$$P \in U \cup V \cup W \implies P \not\equiv P' + P'';$$

And (from $\Theta_n^L(U,V)$*, and the soundness of the proof in which the inference appears) that*

$$a \parallel \Phi_n = \sum U = \sum V = \sum W;$$

Now if $\nexists R \in W$ *st* $R = a \parallel \Phi_n$*, then clearly* $\Theta_n^L(U,W)$;

And if $\exists R \in W$ *st* $R = a \parallel \Phi_n$*, then clearly* $\Theta_n^L(W,V)$;

Similarly, $\Theta_n^L(V,U) \implies \Theta_n^L(W,U) \vee \Theta_n^L(V,W)$;

Hence $\Theta_n(U,V) \implies \Theta_n(U,W) \vee \Theta_n(W,V)$.

The (sub_+) *rule case is similarly straightforward:*

Assume $\Theta_n^L(U \cup U', V \cup V')$;

Again we know immediately that

$$P \in U \cup U' \cup V \cup V' \implies P \not\equiv P' + P'',$$

and

$$\exists P \in U \cup U' \text{ st } P = a \parallel \Phi_n;$$

Suppose this $P \in U$*; then (from* $\Theta_n^L(U \cup U', V \cup V')$*, and the soundness of the proof in which the inference appears) we have that*

$$\sum U = \sum V = a \parallel \Phi_n,$$

so clearly $\Theta_n^L(U,V)$;

And similarly, if this $P \in U'$*, then* $\Theta_n^L(U',V')$.

Similarly, $\Theta_n^L(V \cup V', U \cup U') \implies \Theta_n^L(V,U) \vee \Theta_n^L(V',U')$;

Hence $\Theta_n(U \cup U', V \cup V') \implies \Theta_n(U,V) \vee \Theta_n(U',V')$.

□

6.4.3 Main Result

Here we state and prove our main theorem, the non-finite-axiomatisability of our equivalence.

Theorem 6.4.13 *Suppose that $\mathcal{F}$ is a finite set of (valid) equational axioms. Let n be large enough (as allowed by* **Proposition 6.4.11** *so that no axiom in $\mathcal{F}$ can be instantiated to express any truth $\sum U = \sum V$ where $\Theta_n(U, V)$. Then our natural deduction style equational proof system cannot prove the true statement*

$$a \parallel \Phi_n \;=\; a\Phi_n + a(a \parallel \varphi_1) + a(a \parallel \varphi_2) + \;\cdots\; + a(a \parallel \varphi_n).$$

Therefore no finite complete axiom system can exist for $\mathbf{BPA}_{\parallel}$.

Proof:

Suppose we have a (shortest) proof of the statement

$$a \parallel \Phi_n \;=\; a\Phi_n + a(a \parallel \varphi_1) + a(a \parallel \varphi_2) + \;\cdots\; + a(a \parallel \varphi_n).$$

The proof takes the following form:

$$\frac{\mathcal{D}_0}{\sum U_0 \;=\; \sum V_0}(rule),$$

where

$$U_0 = \left\{a \parallel \Phi_n\right\}$$

and

$$V_0 = \left\{a\Phi_n,\; a(a \parallel \varphi_1),\; a(a \parallel \varphi_2),\; \cdots,\; a(a \parallel \varphi_n)\right\},$$

so clearly $\Theta_n(U_0, V_0)$ holds.

Since this must be a finite *proof, there is somewhere in the proof tree an inference*

$$\frac{\mathcal{D}}{\sum U \;=\; \sum V}(rule) \qquad \textit{where} \qquad \Theta_n(U, V);$$

such that the premise $\mathcal{D}$ of the inference contains no equality

$$\sum U' = \sum V' \qquad \textit{where} \qquad \Theta_n(U', V');$$

By **Proposition 6.4.12**, *(rule) can be neither of (trans) nor (sub_+).*

Furthermore, by **Proposition 6.4.11**, *we know that (rule) cannot be $(t = u)$ for any axiom $t = u \in \mathcal{F}$;*

Also clearly (rule) cannot be (symm), as $\Theta_n(U, V) \iff \Theta_n(V, U)$;

Finally, (rule) cannot be any of $(refl)$, $(sub_\bullet)$, or $(sub_|)$, as this would contradict $\Theta_n(U, V)$;

Hence we have shown that the original statement cannot be proven.

□

Chapter 7

Conclusions and Open Problems

We present here a brief summary of what was accomplished, and what was not accomplished, in the main body of the thesis.

In **Chapter 3**, we investigated axiomatisations of the extensional theories for several languages for nondeterministic and concurrent processes. In that chapter, we were able to present finite ω-complete axiomatisations for a simple language for nondeterminism, and a language with concurrency in the form of the full and left merge operators. However, we failed to find such an axiomatisation, finite or otherwise, for a language with full merge in the absence of the simplifying left merge operator. We do know in fact from a result proven in **Chapter 5** that such an axiomatisation would have to at least be infinite.

In **Chapter 4**, we proved several results on the decomposability of processes into parallel components. In particular, we showed unique factorisation results for our simple concurrent language with and without communication with respect to strong and weak observational congruence. However, in the case of weak observational congruence, the results had to be taken modulo the existence of a τ-factor in the decomposition.

In **Chapter 5**, we proved the nonexistence of a finite axiomatisation for any reasonable congruence for our concurrent language which is at least as strong as strong observational congruence. This result gives us some indication as to where the difficulty arose in **Chapter 3** in trying to ω-completely axiomatise our system,

and furthermore explains the origins of the difficulty faced in trying to completely axiomatise some noninterleaving semantic congruence over this language, when the simple axiom schema given by the Expansion Theorem of **CCS** is not valid.

In **Chapter 6**, we investigated the above questions in the framework of process algebras using sequential composition as opposed to action prefixing. Starting with a model defined by [VRA86], we refined the semantic interpretation and presented a model based on operational semantic methods as in **CCS**. We found a finite complete axiomatisation for the nondeterministic language, but failed to find an ω-complete axiomatisation. With our operational approach, we were able to extend our results on the unique decomposability of processes to this new framework, and prove the nonexistence of a finite axiomatisation for the language with the full merge operator in the absence of the left merge operator.

In the rest of this chapter, we shall outline some of the problems which were not solved in this thesis, but whose solutions would have been included.

7.1 ω-Complete Axiomatisation for the Full Merge Language

In **Chapter 3**, we investigated extensional axiomatisations for several process algebras. We managed to find a relatively easy ω-complete axiomatisation for our language containing both full merge and left merge operators, but we ran upon severe difficulties with the language containing just the full merge operator, in the absence of the simplifying left merge operator. We discovered sequences of independent and arbitrarily-complex Reduction and Absorption Laws which must be accounted for in an ω-complete axiomatisation.

There is very little difference in the above two languages for concurrency. The left merge operator is a slightly restricted version of the full merge operator, which in turn is easily definable in terms of the left merge operator and nondeterministic choice by

$$x \parallel y \;=\; x \lfloor\!\lfloor y \;+\; y \lfloor\!\lfloor x.$$

The complicated axioms discovered in **Section 3.3** are in fact derived from this close relationship between the two algebras. For instance, to derive the Reduction Law

$$\alpha x \parallel (\beta y + z) \;+\; \alpha(x \parallel z) \;=\; \alpha\big(x \parallel (\beta y + z)\big) \;+\; \beta(\alpha x \parallel y) \;+\; \alpha x \parallel z,$$

we take the largest parallel composition, $\alpha x \parallel (\beta y + z)$, which is the term being reduced, and translate it into the left merge language, and then simplify the result using the left merge laws from **Section 3.2** to get

$$\begin{aligned} \alpha x \parallel (\beta y + z) \;&=\; \alpha x \lfloor\!\lfloor (\beta y + z) \;+\; (\beta y + z) \lfloor\!\lfloor \alpha x \\ &=\; \alpha\big(x \parallel (\beta y + z)\big) \;+\; \beta(\alpha x \parallel y) \;+\; z \lfloor\!\lfloor \alpha x. \end{aligned}$$

Adding to both sides of the equation the reverse of the nonsimplifying left merge term $z \lfloor\!\lfloor \alpha x$, namely $\alpha x \lfloor\!\lfloor z = \alpha(x \parallel z)$, and then again simplifying in the obvious way, we arrive at our desired axiom.

Every Reduction Law introduced in **Section 3.3** can be viewed, and indeed was first conceived of, in the above manner. By translating a term into the left merge language and then distributing the left merge operator through the nondeterministic choice from the right, we can express all of the possible first-step behaviours of the process term separately. This gives us the minimal amount of first-step behavioural properties which must exist in a process term which represents an expression containing our reduced-out term as a summand. This is the sense in which we argued in **Section 3.3** that our axioms were minimal. Furthermore, whenever some term $x \lfloor\!\lfloor t$ appears as a summand in this minimal first-step behaviour expression, we know that the process represented by the term must have the capability of proceeding via the indeterminate process x in the context where it is running in parallel with the term t. Thus we know that the behaviour $t \lfloor\!\lfloor x$ must be present in the minimal first-step behaviour expression. The Absorption Laws were originally conceived of in an identical fashion.

Using this as a guide, we can attempt to prove the ω-completeness of the infinite set of axioms presented in **Section 3.3** including the sequences of Reduction and Absorption Laws. That is, we can perhaps utilise the left merge language, for which we have complete knowledge of the extensional theory, as a metalanguage for defining a normal form, and use properties derived from considerations of the special properties of full merge to arrive at our completeness result. For instance, we can start by using the left merge denotation function to distinguish terms. That is, we have the denotation function $[\![\cdot]\!]$ from **Definition 3.2.5** restricted to terms in the sublanguage $\mathcal{P}_2$ of $\mathcal{P}_1$ where the left merge operator does not appear. Recall that this denotation maps elements of $\mathcal{P}_1$ (and hence elements of $\mathcal{P}_2$ as well) to the domain $\mathcal{D}_1$ given as the least fixed-point solution to the set-theoretic equation

$$\mathcal{D}_1 \;=\; \mathcal{P}_{FIN}(\mathbf{Var} \times \mathcal{D}_1 \;\cup\; \mathbf{Act} \times \mathcal{D}_1),$$

where $\mathcal{P}_{FIN}(S)$ represents the set of finite subsets of S. The denotation of a term gives the set of action–derivative pairs specifying what initial actions are possible along with the resulting derived terms which the process would evolve into, and the variable–process pairs specifying what indeterminate processes appear unguarded in the term along with the processes representing the parallel context within which the variable processes appear. For instance, the term $a + b \parallel x$ would have a denotation representing the normal form $a + bx + x \lfloor\!\lfloor b$.

We know immediately that this denotation function is consistent with respect to our equational theory. That is, due to the soundness of the axioms, we know that whenever the axioms can prove $t = u$, we have that $[\![t]\!] = [\![u]\!]$. The problem here is to show the reverse implication. The difficulty lies in that not every element of the domain $\mathcal{D}_1$ is the image of a term in $\mathcal{P}_2$. For instance, the domain element $\{(x, a)\}$ represents the left merge term $x \lfloor\!\lfloor a$ which has no equivalent form in the subalgebra $\mathcal{P}_2$. Thus we need to place restrictions on the possible forms of domain elements which can be the images of $\mathcal{P}_2$ terms. With the motivation forming the basis of our axiom construction above, we can immediately come up with the

following conjecture on a restriction for the denotations of terms from the full merge language.

Conjecture 7.1.1 *For terms* $t, t' \in \mathcal{P}_2$, $(x, [\![t']\!]) \in [\![t]\!]$ *implies* $[\![t' \lfloor\!\lfloor (x + t'')]\!] \subseteq [\![t]\!]$ *for some* $t'' \in \mathcal{P}_2$.

What this conjecture is telling us is that whenever within a term t, we can proceed with an indeterminate process x within the parallel context $(x + t'') \parallel t'$, then we must also be able to proceed in that context with any initial transition offered by t'. This is reflected in our (minimal) Reduction Law construction method above, where we add the term $t' \lfloor\!\lfloor x$ to both sides of our equation when one side has the nonsimplifying term $x \lfloor\!\lfloor t'$ as a summand.

We can then proceed to define the normal form of a term $t \in \mathcal{P}_1$ by

$$nf(t) \;=\; \sigma([\![t]\!]),$$

where σ would be defined somehow like the following:

$$\sigma(T) \;=\; \sum_{(a,S)\in T} a.\sigma(S) \;+\; \sum_{(x,S)\in T} x \parallel \sigma(S).$$

The only difference between this definition and the one of **Definition 3.2.10** is that here we have $x \parallel t$ instead of $x \lfloor\!\lfloor t$. This is not a correct definition, as it fails to take account of the extra t'' term from **Conjecture 7.1.1**. How to define that extra term is where we find the first problem to this approach. However, given that we could repair the definition, with the help of **Conjecture 7.1.1**, we would next like to prove the following conjecture.

Conjecture 7.1.2 *With our axioms, we can prove the statement* $t = nf(t)$.

This would be where our greatest problems would lie, as it is where we must make our axioms work for us. The difficulty would arise in a large part due to the indeterminate nature of the t'' in **Conjecture 7.1.1**. However, assuming it to be

true, we would have as an immediate corollary that we could prove the statement $t = u$ with our axiom set whenever $[\![t]\!] = [\![u]\!]$.

From here it would just remain to show that the denotation is sound with respect to our congruence, that is, that $[\![t]\!] = [\![u]\!]$ whenever $t \sim u$. However, we get this result almost for free from **Proposition 3.2.14** and **Corollary 3.2.15.**

Hence, this approach to the solution to our ω-completeness problem looks very promising. However, we have not been able yet to follow out the details to a successful end.

7.2 The Non-Finite-Axiomatisability of Observational Congruence

In **Chapter 5**, we managed to prove the nonexistence of any finite axiomatisation first for strong congruence, and then for any reasonable congruence at least as strong as strong congruence, over the languages $\mathcal{P}_2^0$ and $\mathcal{P}_3^0$, the languages containing full merge and merge with synchronisation respectively in their signatures. As we recall, the usual complete axiomatisation for strong congruence $\sim$ over $\mathcal{P}_3^0$ is given by $\mathcal{T}_{str}$ consisting of the following axioms.

$$(A_1)\ \ (x+y)+z = x+(y+z); \qquad (A_3)\ \ x+x = x;$$
$$(A_2)\ \ x+y = y+x; \qquad (A_4)\ \ x+0 = x;$$

$$(Exp_{mn})\ \textit{For}\ u = \sum_{i=1}^{m} \mu_i x_i\ \textit{and}\ v = \sum_{j=1}^{n} \nu_j y_j,$$

$$u \mid v = \sum_{i=1}^{m} \mu_i (x_i \mid v) + \sum_{j=1}^{n} \nu_j (u \mid y_j) + \sum_{\mu_i = \bar{\nu}_j} \tau (x_i \mid y_j).$$

What we did not do, and which is sorely lacking in the chapter, is prove an analogous result for the weaker observational congruence over these languages. Recall that the usual complete axiomatisation for observational congruence $\approx^c$ over $\mathcal{P}_3^0$ is given by $\mathcal{T}_{obs}$ consisting of the axioms of $\mathcal{T}_{str}$ above, along with the following τ laws.

$$(T_1)\quad \mu\tau x = \mu x;$$
$$(T_2)\quad x + \tau x = \tau x;$$
$$(T_3)\quad \mu(x + \tau y) + \mu y = \mu(x + \tau y).$$

Though the conjecture is that these systems are equally not finitely axiomatisable, the proof technique used for the previous results falls through in the presence of silent τ actions.

One immediate problem is that we no longer have the property that congruence respects syntactic depth. For instance,

$$a.\tau.t \approx^c a.t,$$

whereas

$$|a.\tau.t| = 2 + |t| \neq 1 + |t| = |a.t|.$$

However, this particular problem can be easily remedied by redefining the depth function $|\cdot|$ to allow it to ignore τ actions. We can then show that observational congruence, as indeed observational equivalence, respects this new definition.

Our bigger problem arises with actually arguing about terms derived from congruent terms, which can be assumed to be observationally equivalent, but not necessarily congruent. Furthermore, we cannot easily manipulate syntactic terms which derive under an indeterminate number of silent τ actions into some term equivalent to a behaviour in which we are interested. In relation to our proof in **Chapter 4** on the unique decomposition of agents in $\mathcal{P}_3^0$ with respect to observational congruence, we cannot assume given a proof tree of a statement that all subterms which appear in the tree are going to be expressible as proper normal forms, even when the statement which is being proven by the proof tree expresses a relationship between proper normal form terms. In particular, we would still like to prove that all statements of the form

$$a \mid (a + aa + \cdots + a^n)$$
$$= a(a + aa + \cdots + a^n) + aa + aaa + \cdots + a^{n+1}$$

cannot be proven using any finite set of valid equational axioms.

Ideally, we would like to show some relationship between strong congruence and observational congruence which would allow us to infer that given a finite axiomatisation for the latter, we could deduce a finite axiomatisation for the former. Given this, our problem would be solved, as we have already shown the non-finite-axiomatisability of strong congruence. Notice that the reverse direction can be seen to hold: given a (finite) axiomatisation for strong congruence, we would simply need to add to this axiomatisation the three τ laws $\{(T_1), (T_2), (T_3)\}$ from above in order to get a (finite) axiomatisation for observational congruence.

Unfortunately, such a relationship is not obvious. One may think it sufficient to remove any and all laws which manipulate silent τ actions from an axiomatisation for observational congruence in order to derive a (smaller) set of axioms for strong congruence. However, we can see that such an approach fails immediately, by noting that $\mathcal{T}_{obs} \backslash \{(A_3)\}$ is a complete axiomatisation for observational congruence, all ground instances of the idempotence of $+$ being derivable using the other laws. For example,

$$\begin{aligned}
a + a &= a\tau\tau + a\tau && \textit{(using } (T_1) \textit{ three times)} \\
&= a(\tau + \tau\tau) + a\tau && \textit{(using } (T_2)\textit{)} \\
&= a(\tau + \tau\tau) && \textit{(using } (T_3)\textit{)} \\
&= a\tau\tau && \textit{(using } (T_2)\textit{)} \\
&= a. && \textit{(using } (T_1) \textit{ twice)}
\end{aligned}$$

However, by removing the τ laws from $\mathcal{T}_{obs} \backslash \{(A_3)\}$, we would arrive at $\mathcal{T}_{str} \backslash \{(A_3)\}$, in which we could not prove that $a + a = a$.

An equally ideal situation would be to reason that given a finite complete axiomatisation for $\approx^c$, we could add a finite number of sound laws in order to derive a modified axiomatisation in which we could prove any statement $p = q$ in which p and q only contain the action a in their sorts ($\mathrm{Sort}(p) = \mathrm{Sort}(q) = \{a\}$) without using any term possessing a sort different from $\{a\}$. With this, our analysis could mimic the strong congruence case (we would no longer need to worry about silent τ

transitions, as no τ actions would appear explicitly, nor could any communications occur, given the further assumption that $\bar{a} \neq a$).

An analogous trick was used in **Section 5.1** to eliminate any unnecessary $\mathbf{0}$'s appearing as subterms in proofs. We started with a (supposedly) finite, complete and sound set of equational axioms for $\sim$, and added to these a finite number of additional axioms which allowed us to prove any valid statement not containing any $\mathbf{0}$ summands or factors in either agent using only similar agents in the proof tree. In this case, we would like to assume given a finite set F of sound and complete laws for $\approx^c$, and extend this to some superset $\mathcal{F}$ of sound laws by adding a finite collection of new laws, so that we can guarantee that there will be proof trees for the valid statements

$$a \mid (a + aa + \cdots + a^n)$$

$$= \; a(a + aa + \cdots + a^n) \; + \; aa \; + \; aaa \; + \; \cdots \; + \; a^{n+1},$$

in which every term P appearing in the proof trees satisfy $\mathrm{Sort}(P) = \{a\}$.

Following the technique employed in **Section 5.1**, what we would like to deduce is that any proof of a statement $P = Q$ in a system parameterised by a finite and complete set of axioms F, where P and Q each have sort $\{a\}$, could be replaced by a valid proof of the same statement over the extended set $\mathcal{F}$ of laws by replacing each inference

$$\frac{\cdots \; p_i = q_i \; \cdots}{p = q}(rule)$$

by an inference

$$\frac{\cdots \; \widehat{p_i} = \widehat{q_i} \; \cdots}{\widehat{p} = \widehat{q}}(rule'),$$

where $\widehat{p}$ is p with all τ actions removed. Formally,

$$\widehat{\mathbf{0}} \; = \; \mathbf{0}; \qquad\qquad \widehat{p + q} \; = \; \widehat{p} + \widehat{q};$$

$$\widehat{\mu p} \; = \; \begin{cases} \mu\widehat{p}, & \text{if } \mu \neq \tau, \\ \widehat{p}, & \text{if } \mu = \tau; \end{cases} \qquad\qquad \widehat{p \mid q} \; = \; \widehat{p} \mid \widehat{q}.$$

Bibliography

[AUS84] Austry, D., G. Boudol, *"Algèbre de Processus et Synchronisation"*, Theoretical Computer Science, Vol 30, No 1, 1984.

[BAR84] Barendregt, H.P., *The Lambda Calculus — Its Syntax and Semantics, Studies In Logic and The Foundations of Mathematics*, Vol 103, North-Holland, 1984.

[BER84] Bergstra, J.A., J.W. Klop, *"Process Algebra for Synchronous Communication"*, Information and Computation, Volume 60, Number 1/3, 1984.

[BER85] Bergstra, J.A., J.W. Klop, *"Algebra of Communicating Processes with Abstraction"*, Theoretical Computer Science, Vol 37, No 1, 1985.

[BOU85] Boudol, G., *"Notes on Algebraic Calculi of Processes"*, Logics and Models of Concurrent Systems, NATO ASI Series F13, Springer–Verlag, 1985.

[BOU86] Boudol, G., I. Castellani, *"On the Semantics of Concurrency: Partial Orders and Transition Systems"*, Proceedings of TAPSOFT '87, Vol I, Lecture Notes in Computer Science 249, Springer–Verlag, 1987.

[BRO84] Brookes, S.D., C.A.R. Hoare, A.W. Roscoe, *"A Theory of Communicating Sequential Processes"*, Journal of the ACM, Vol 31, No 3, 1984.

[CAS87] Castellani, I., M. Hennessy, *"Distributed Bisimulation"*, University of Sussex Computer Science Report No. 5/87, July 1987.

[CAS88] Castellani, I., *"Bisimulations for Concurrency"*, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, Report Number CST-51-88, 1988.

[CON71] Conway, J.H., *Regular Algebra and Finite Machines*, Chapman and Hall, London, 1971.

[CRA58] Craig, W., R. Vaught, *"Finite Axiomatisability Using Additional Predicates"* Journal of Symbolic Logic, Vol 23, No 3, 1958.

[DEN84] De Nicola, R., M.C.B. Hennessy, *"Testing Equivalence for Processes"*, Theoretical Computer Science, Vol 34, No 1/2, 1984.

[EHR85] Ehrig, H., B. Mahr, *Fundamentals of Algebraic Specification 1 — Equations and Initial Semantics*, Springer–Verlag, 1985.

[GIS84] Gischer, J.L., *"Partial Orders and the Axiomatic Theory of Shuffle"*, Ph.D. Thesis, Stanford University, Report No. STAN-CS-84-1033, December 1984.

[HEE86] Heering, J., *"Partial Evaluation and ω-completeness of Algebraic Specifications"*, Theoretical Computer Science, Vol 43, 1986.

[HEN85] Hennessy, M., R. Milner, *"Algebraic Laws for Nondeterminism and Concurrency"*, Journal of the ACM, Vol 32, No 1, 1985.

[HEN87] Hennessy, M., *"Axiomatising Finite Concurrent Processes"*, University of Sussex Computer Science Report No. 4/87, July 1987.

[HIN86] Hindley, J.R., J.P. Seldin, *Introduction to Combinators and λ-Calculus*, Cambridge University Press, 1986.

[HOA78] Hoare, C.A.R., *"Communicating Sequential Processes"*, Communications of the ACM, Vol 21, No 8, 1978.

[HOA85] Hoare, C.A.R., *Communicating Sequential Processes*, Prentice-Hall International, 1985.

[KEL76] Keller, R., *"Formal Verification of Parallel Programs"*, Communications of the ACM, Vol 19, No 7, 1976.

[KLE52] Kleene, S.C., *"Theories in the Predicate Calculus Using Additional Predicate Symbols"*, Memoirs of the American Mathematical Society, 1952.

[KLO87] Klop, J.W., Personal Communication, October, 1987.

[MIL80] Milner, R., *A Calculus of Communicating Systems*, Lecture Notes in Computer Science 92, Springer–Verlag, 1980.

[MIL84] Milner, R., *"A Complete Inference System for a Class of Regular Behaviours"*, Journal of Computer and System Sciences, Vol 28, No 3, 1984.

[MIL85] Milner, R., *"Lectures on a Calculus for Communicating Systems"*, Lecture Notes in Computer Science 197, Springer–Verlag, 1985.

[MIL86] Milner, R., *"A Complete Axiomatisation for Observational Congruence of Finite-state Behaviours"*, Laboratory for the Foundations of Computer Science, Department of Computer Science, University of Edinburgh Research Report ECS-LFCS-86-8, 1986.

[NIE81] Nielson, M., G. Plotkin, G. Winskel, *"Petri Nets, Event Structures and Domains, Part I"*, Theoretical Computer Science, Vol 13, No 1, 1981.

[PAR81] Park, D.M.R., *"Concurrency and Automata on Infinite Sequences"*, Proceedings of the 5^{th} G.I. Conference, Lecture Notes in Computer Science 104, Springer–Verlag, 1981.

[PLO81] Plotkin, G., *"A Structured Approach to Operational Semantics"*, DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[PRA65] Prawitz, D., *Natural Deduction*, Almqvist and Wiksell, Stockholm, 1965.

[PRA86] Pratt, V., *"Modeling Concurrency with Partial Orders"*, International Journal of Parallel Programming, Vol 15, No 1, 1986.

[SAN82] Sanderson, M.T., *"Proof Techniques for* **CCS***"*, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, Report Number CST-19-82, 1982.

[SHO67] Shoenfeld, J.R., *Mathematical Logic*, Addison–Wesley, 1967.

[VRA86] Vrancken, J.L.M., *"The Algebra of Communicating Processes with Empty Process"*, Department of Computer Science Report FVI 86-01, University of Amsterdam, 1986.

[WIN80] Winskel, G., *"Events in Computation"*, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, Report Number CST-10-80, 1980.

[WIN83] Winskel, G., *"Event Structure Semantics for CCS and Related Languages"*, Department of Computer Science Report DAIMI PB-159, Aarhus University, 1983.