A Semantic Universe for the study of Fairness

VERY ROUGH AND INCOMPLETE DRAFT

Peter Aczel (petera@cs.man.ac.uk)
Departments of Mathematics and Computer Science
Manchester University
Manchester M13 9PL, U.K.

October 1, 1996

1 Introduction

I give an application of the general final universe approach to semantics, of [FUP], to give a new semantics for the study of fairness. My starting point is the approach to fairness in $SCCS$ initiated by Milner in [Miln], where he added to $SCCS$ a finite delay operator. Recently, in [Hart1], Milner's operational semantics for finite delay was developed into a final universe semantics and this universe was used in [Hart2] to give the semantics of a variant of $SCCS$ in which the fixpoint operators are refined into two kinds of fixpoint operators and in which the finite delay operator can be represented.

In this paper I propose a much richer final universe for the study of fairness than that used in [Hart1] and [Hart2]. In my view the semantics to finite delay, initiated in [Miln], is too course to make intuitively desirable distinctions, so that something like the present proposal is needed. It should be noted that the idea behind my proposal may already be seen in [Henn1].

The operational semantics of $SCCS$ involves a labelled transition relation giving transition steps

$$p \xrightarrow{a} q$$

between agents p, q, labelled with an atomic action $a \in Act$. A computation

$$p_0 \xrightarrow{a_0} p_1 \xrightarrow{a_1} \cdots$$

consists of a sequence of such transition steps. The topic of fairness arises when, for various reasons, certain infinite computations are considered unfair and therefore to be excluded from consideration. In [Miln] Milner introduces a simple setting for the study of fairness. The idea is to add to $SCCS$ a finite delay operator ϵ. So, for each agent p there is an agent ϵp. This agent can behave like p, so that there is a transition $\epsilon p \xrightarrow{a} q$ whenever p has the transition $p \xrightarrow{a} q$. But it can also delay behaving like p because there is a transition step $\epsilon p \xrightarrow{1} \epsilon p$, where $1 \in Act$ is a special atomic action representing an atomic delay. While ϵp can repeatedly delay behaving like p it must eventually behave like p, so that the infinite computation

$$\epsilon p \xrightarrow{1} \epsilon p \xrightarrow{1} \cdots$$

is not allowed unless there is a computation step $p \xrightarrow{1} \epsilon p$. This infinite computation can be considered unfair when ϵp is acting in parallel with an agent q that is waiting for an action of p, which is forever delayed.

The approach of [Miln] to the semantics of finite delay, that is developed further in [Hart1] and [Hart2] is to associate with each agent p a set $V(p)$ of those infinite sequences of atomic actions that label the allowed infinite computations of p. In the case of the finite delay operator we get

$$V(\epsilon p) = \{1^n \sigma \mid n \in \omega \;\&\; \sigma \in V(p)\}.$$

Here, if $\sigma \in Act^\omega$ is the infinite sequence $(a_0, a_1, \ldots)$, with $a_0, a_1 \ldots \in Act$ then $1^n\sigma$ is the infinite sequence obtained from σ by prefixing it with n 1s. Note that the infinite sequence 1^ω, consisting only of 1s, will only be in $V(\epsilon p)$ if it is already in $V(p)$.

The operator ϵ needs to be contrasted with an operator δ that can already be defined in $SCCS$ using the recursion

$$\delta p = \mathsf{fix} x(1.x + p).$$

so that

$$\delta p = 1.\delta p + p.$$

In fact we also have

$$\epsilon p = 1.\epsilon p + p.$$

But, in contrast to ϵp the agent δp allows the infinite computation

$$\delta p \xrightarrow{1} \delta p \xrightarrow{1} \cdots$$

so that

$$V(\delta p) = V(\epsilon p) \cup \{1^\omega\}.$$

So, when $1^\omega \in V(p)$ then $V(\epsilon p) = V(\delta p)$ so that, for Milner, $\epsilon p = \delta p$. But when $1^\omega \notin V(p)$ then $V(\epsilon p) \neq V(\delta p)$, so that for Milner $\epsilon p \neq \delta p$.

Now consider the following example, where $a \in Act$ is distinct from 1. Let $p = 1.\delta(a.\mathbf{0})$, where $\mathbf{0}$ is the null agent that allows no transitions at all. Using the obvious equations

$$V(\mathbf{0}) = \emptyset$$

and

$$V(a.p) = \{a\sigma \mid \sigma \in V(p)\},$$

we easily get that $1^\omega \in V(p)$ so that $\epsilon p = \delta p$.

But I claim that intuitively the agents ϵp and δp behave differently. This is because ϵp must eventually allow an a-transition, but δp may never allow an a-transition by delaying for ever.

The problem is that the semantics in [Miln] focusses on the allowed infinite sequences of atomic actions, abstracting away from the intermediate agents that an infinite computation passes through. In this respect the semantics is related to the trace semantics which is only concerned with the allowed finite sequences of atomic actions, again abstracting away from the agents a finite computation passes through. In fact it is Milner's strong bisimulation equivalence which fully takes into account the intermediate agents of a finite computation. The strong bisimulation idea is indeed used in [Miln] for finite computations, but infinite traces are used for infinite computations. My proposal is to discard this hybrid approach in favour of a full bisimulation approach for both finite and infinite computations.[1]

2 Review of $SCCS$ syntax and operational semantics, with the resulting notions of computation and bisimulation

We start with a review of the version $SCCS(K, \kappa)$ of $SCCS$ that uses a class K of constants with possibly recursive defining equations, given by κ. The version $SCCS_\mu$ that uses fixed point operators that bind agent variables will need to be treated later.

2.1 Syntax

We assume given an Abelian group $(Act, \times, 1)$ of atomic actions, a. For each class K of **constants**, c, we define a class E_K of **agents (agent expressions)**, e, using the following BNF style syntax equation.

$$\boxed{e ::= \quad c \quad | \quad a.e \quad | \quad \Sigma\tilde{e} \quad | \quad e_1 \times e_2 \quad | \quad e\lceil A,}$$

where $\tilde{e}$ can be any family $\{e_i\}_{i \in I}$ of agents e_i, for any index set I and $A \subseteq Act$.

2.2 The Operational Semantics

The operational semantics is given relative to a choice of a **defining equation**

$$c =_{def} e_c,$$

for each $c \in K$, where $e_c \in E_K$. This choice is given by the function $\kappa : K \to E_K$, where $\kappa(c) = e_c$ for $c \in K$. Given (K, κ) the $SCCS(K, \kappa)$ operational semantics consists of the following clauses of an inductive definition of transition relations $\xrightarrow{a}$ on E_K for $a \in Act$.

[1] Sept 30, 1996: At CSL96 last week Colin Stirling informed me that Milner himself had preferred the full bisimulation notion, but worked with the hybrid notion in his 1982 report because of technical difficulties with the full bisimulation notion. Colin also drew my attention to: "The Power of the Future Perfect in Program Logics", by Matthew Hennessy and Colin Stirling. I believe this is in Information and Control 1984. There is an Edinburgh Technical Report, CSR-156-83, (December, 1983). That paper works with the full bisimulation notion for transition systems with a general notion of admissible infinite computation, as here.

$$
a.e \xrightarrow{a} e \qquad \frac{e_i \xrightarrow{a} e}{\Sigma\tilde{e} \xrightarrow{a} e}(i \in I) \qquad \frac{e_c \xrightarrow{a} e}{c \xrightarrow{a} e}(c \in K)
$$

$$
\frac{e \xrightarrow{a} e'}{e\lceil A \xrightarrow{a} e'\lceil A}(a \in A) \qquad \frac{e_1 \xrightarrow{a_1} e_1' \quad e_2 \xrightarrow{a_2} e_2'}{e_1 \times e_2 \xrightarrow{a} e_1' \times e_2'}(a = a_1 \times a_2)
$$

2.3 Computations

The transition rules determine the computations. An $SCCS(K, \kappa)$**-computation from an agent** e has the form $\rho = e \xrightarrow{a_0} e_0 \xrightarrow{a_1} e_1 \cdots$ where $(a_0, e_0)(a_1, e_1)\cdots$ is a finite or infinite sequence of action-agent pairs such that $e \xrightarrow{a_0} e_0$, $e_0 \xrightarrow{a_1} e_1, \cdots$. Its **length**, $len(\rho)$, is the length of the sequence of action-agent pairs and so is a natural number for finite sequences, or ∞ for infinite sequences. We allow the empty sequence of action-agent pairs and call the resulting computation of length 0 the **trivial** computation from e.

Let $\rho = e \xrightarrow{a_0} e_0 \xrightarrow{a_1} e_1 \xrightarrow{a_2} \cdots$ be a computation from e. Define $act(\rho) = \{a_0, a_1, a_2, \ldots\}$ and, when ρ is non-trivial let ρ^+ be the computation

$$
\rho^+ = e_0 \xrightarrow{a_1} e_1 \xrightarrow{a_2} \cdots.
$$

We write $\rho = e \xrightarrow{a_0} \rho^+$. For each $A \subseteq Act$, if $act(\rho) \subseteq A$ then let

$$
\rho\lceil A = e\lceil A \xrightarrow{a_0} e_0\lceil A \xrightarrow{a_1} e_1\lceil A \xrightarrow{a_2} \cdots.
$$

This is a computation from $e\lceil A$. Also if, for $i = 1, 2$

$$
\rho^i = e^i \xrightarrow{a_0^i} e_0^i \xrightarrow{a_1^i} e_1^i \xrightarrow{a_2^i} \cdots
$$

is a computation from e^i, with ρ^1, ρ^2 of the same length, then let

$$
\rho^1 \times \rho^2 = e \xrightarrow{a_0} e_0 \xrightarrow{a_1} e_1 \xrightarrow{a_2} \cdots
$$

where, $e = e^1 \times e^2$ and for $n = 0, 1, \ldots$, $e_n = e_n^1 \times e_n^2$ and $a_n = a_n^1 \times a_n^2$. This is a computation from e.

For any relation R on agents we may lift the relation to a relation on computations as follows. If $\rho^i = e^i \xrightarrow{a_0^i} e_0^i \xrightarrow{a_1^i} e_1^i \xrightarrow{a_2^i} \cdots$ is a computation from e^i for $i = 1, 2$ then we write $\rho^1 R \rho^2$ if ρ^1, ρ^2 have the same length and $e^1 R e^2$, $a_0^1 = a_0^2$, $e_0^1 R e_0^2$, $a_1^1 = a_1^2$, ...

Proposition: 2.1

1. *Any computation from an agent $e\lceil A$ has the form $\rho\lceil A$ for some computation ρ from e whose actions are in A.*

2. *Any computation from an agent $e^1 \times e^2$ has the form $\rho^1 \times \rho^2$ for some computations ρ^1, ρ^2 from e^1, e^2, respectively, of the same length.*

2.4 Bisimulation

Given the transition relations we get the notion of $SCCS(K, \kappa)$-bisimulation on E_K.

Definition: 2.2 *A relation R on E_K is a* **simulation** *if, whenever $eRf \xrightarrow{a} f'$ then there is e' such that $e \xrightarrow{a} e'Rf'$. The relation R is a* **bisimulation** *if both R and its converse R^{-1} are simulations.*

Proposition: 2.3 *There is a maximal bisimulation $\sim$. Moreover $\sim$ is an equivalence relation on E_K and is a congruence with respect to the SCCS combinators, action prefix, summation, product and restriction.*

If we lift $\sim$ to computations we can state the following result.

Proposition: 2.4 *If $e^1 \sim e^2$ and ρ^2 is a computation from e^2 then there is a computation ρ^1 from e^1 such that $\rho^1 \sim \rho^2$.*

2.5 The version $SCCS_\mu$ of $SCCS$

This is the version that uses an unlimited supply[2] of **agent variables**, x, and for each non-repeating family $\tilde{x} = \{x_i\}_{i \in I}$ of variables, indexed by an index set I, and $i \in I$ a variable binding fixed point operator $\mu_i \tilde{x}$. This operator can be applied to any I-indexed family $\tilde{e} = \{e_i\}_{i \in I}$ of agent expressions to obtain an agent expression $\mu_i \tilde{x}\tilde{e}$. In the usual way we have the notions of free and bound occurrences of variables. In particular free occurrences of any agent variable from $\tilde{x}$ in any agent expression from $\tilde{e}$ become bound in $\mu_i \tilde{x}\tilde{e}$. An agent expression is an **agent of** $SCCS_\mu$ if it has no free occurrences of variables.

We shall present the operational semantics of $SCCS_\mu$ by representing $SCCS_\mu$ as a special case $SCCS(K_\mu, \kappa_\mu)$ of the previous approach to $SCCS$. We treat the μ-expressions $\mu_i \tilde{x}\tilde{e}$ that are agents as the constants forming the class K_μ. Note that the constants are no longer primitive, but this is not a problem. The important thing is that the constant agents are distinguishable from the agents built up using the standard $SCCS$-combinators. The class of $SCCS_\mu$-agents now becomes the class E_{K_μ}. It remains to specify the function κ_μ that determines the defining equations for the constants.

If $c = \mu_i \tilde{x}\tilde{e} \in K_\mu$ is a constant then its defining equation $c =_{def} e_c$ is determined by letting

$$e_c = e_i[\tilde{\mu}\tilde{x}\tilde{e}/\tilde{x}];$$

i.e. e_c is the result of simultaneously substituting the elements of the family $\tilde{\mu}\tilde{x}\tilde{e} = \{\mu_i \tilde{x}\tilde{e}\}_{i \in I}$ for the corresponding variable in $\tilde{x}$ in the agent expression e_i. So we let $\kappa_\mu(c) = e_c$.

We have represented the syntax of $SCCS_\mu$ as that of $SCCS(K_\mu, \kappa_\mu)$ thereby determining the transition relations and computations and the notion of bisimulation.

[2]Actually, a proper class is needed

3 $SCCS$ with a notion of admissible computation

We refine $SCCS(K,\kappa)$ by adding a notion of 'admissible infinite computation'. We do this by postulating a function ad that assigns to each agent e a set $ad(e)$ of the **admissible** computations from e. Of course every admissible computation from e must be a computation from e. We require the following additional **admissibility conditions** to hold.

1. The trivial computation from any agent is admissible.
2. A non-trivial computation ρ is admissible if and only if ρ^+ is admissible.
3. If ρ is a computation from e, all of whose actions are in $A \subseteq Act$, then ρ is admissible if and only if $\rho\lceil A$ is admissible; i.e.
$$ad(e\lceil A) = \{\rho\lceil A \mid \rho \in ad(e) \ \ \& \ \ act(\rho) \subseteq A\}.$$
4. Let ρ be a computation from an agent $e^1 \times e^2$. Then ρ is admissible if and only if there are admissible computations ρ^1 from e^1 and ρ^2 from e^2, of the same length as that of ρ, such that $\rho = \rho^1 \times \rho^2$; i.e.
$$ad(e^1 \times e^2) = \{\rho^1 \times \rho^2 \mid \rho^1 \in ad(e^1) \ \ \& \ \ \rho^2 \in ad(e^2) \ \ \& \ \ len(\rho^1) = len(\rho^2)\}.$$

It easily follows from the first two conditions that

Proposition: 3.1

1. *Every finite computation is admissible.*
2. $ad(a.e) = \{a.e \xrightarrow{a} \rho \mid \rho \in ad(e)\}$.
3. $ad(\Sigma\tilde{e}) = \{\Sigma\tilde{e} \xrightarrow{a} \rho \mid e_i \xrightarrow{a} \rho \in ad(e_i) \textit{ for some } i \in I\}$.
4. $ad(c) = \{c \xrightarrow{a} \rho \mid e_c \xrightarrow{a} \rho \in ad(e_c)\}$.

We now need to formulate a modified notion of bisimulation that takes account of ad.

3.1 ad-Bisimulations

Definition: 3.2 *A relation R between agents is an ad-***simulation** *if,*
$$e^1 R e^2 \ \ \Rightarrow \ \ \forall \rho^2 \in ad(e^2)\, \exists \rho^1 \in ad(e^1) \ \ \rho^1 R \rho^2.$$
*R is an ad-***bisimulation** *if both R and its converse are ad-simulations.*

Proposition: 3.3 *There is a maximal ad-bisimulation $\sim_{ad}$. Moreover $\sim_{ad}$ is an equivalence relation that is a congruence with respect to the $SCCS$ combinators prefix, summation, product and restriction.*

3.2 The T-admissible infinite computations

The admissible infinite computations will be specified relative to a stable class, T, of infinite sequences of elements of K. The stability of T means that an infinite sequence p of constants is in T if and only if the sequence p^+, obtained from p by dropping the first term, is in T.

We start by giving an analysis of the transition steps. Let D be the class of **special** agents, d, that are given by

$$\boxed{d ::= \quad c \mid \ a.e \ \mid \ d\lceil A \ \mid \ d \times d}$$

and inductively define the relation $\triangleright$ using the following clauses.

$$\boxed{\begin{array}{cc} d \triangleright d \ \ (d \in D) & \dfrac{e_i \triangleright d}{\Sigma\tilde{e} \triangleright d}\ (i \in I) \\[2ex] \dfrac{e_1 \triangleright d_1 \quad e_2 \triangleright d_2}{e_1 \times e_2 \ \triangleright \ d_1 \times d_2} & \dfrac{e \triangleright d}{e\lceil A \triangleright d\lceil A}\ (A \subseteq Act) \end{array}}$$

Note that if $e \triangleright d$ then d is special.

Proposition: 3.4

$$e \xrightarrow{a} e' \iff e \triangleright d \xrightarrow{a} e' \textit{ for some } d.$$

We now want to associate with each $d \in D$ a finite set $P(d) \subseteq \{1,2\}^*$ of **positions** of d and then define for each position $\sigma \in P(d)$ an element $d^\sigma \in K_\Box = K \cup \{\Box\}$, where $\Box \notin K$. We use ϵ for the empty string.

$$\begin{cases} P(a.e) & = \{\epsilon\} \\ P(c) & = \{\epsilon\} \\ P(d\lceil A) & = P(d) \\ P(d_1 \times d_2) & = \{i\sigma \mid i \in \{1,2\} \ \ \& \ \ \sigma \in P(d_i)\} \end{cases}$$

$$\begin{cases} (a.e)^\epsilon & = \Box \\ c^\epsilon & = c \\ (d\lceil A)^\sigma & = d^\sigma \\ (d_1 \times d_2)^{i\sigma} & = d_i^\sigma \end{cases}$$

Use $\leq$ for the prefix relation on strings and define

$$d \to d' \iff \forall \sigma \in P(d)\, \exists \sigma' \in P(d') \ \ \sigma \leq \sigma' \ \ \& \ \ \forall \sigma' \in P(d') \exists \sigma \in P(d) \ \ \sigma \leq \sigma'.$$

Call $\sigma_0\sigma_1 \cdots$ a **calling path** for $d_0 d_1 \cdots \in D^\omega$ if for all $n \in \mathbb{N}$

$$\sigma_n \in P(d_n) \ \ \& \ \ \sigma_n \leq \sigma_{n+1}.$$

Definition: 3.5 *An infinite computation* $e \xrightarrow{a_0} e_0 \xrightarrow{a_1} e_1 \cdots$ *is* T**-admissible** *if there is* $d_0 d_1 \cdots \in D^\omega$ *such that*

$$e \triangleright d_0 \xrightarrow{a_0} e_0 \triangleright d_1 \xrightarrow{a_1} \cdots$$

and for every calling path $\sigma_0\sigma_1 \cdots$ *for* $d_0 d_1 \cdots \in D^\omega$

$$d_0^{\sigma_0} d_1^{\sigma_1} \cdots \in T_\Box,$$

where $T_\Box$ *is the class of all infinite sequences of elements of* $K_\Box$ *whose restriction to K is in* T.

3.3 $!SCCS_\mu$

For $!SCCS_\mu$ we refine the fixed point operators $\mu_i\tilde{x}$ of $SCCS_\mu$ to operators $\mu_i^S\tilde{x}$, where S must be a set of infinite sequences of elements of the index set I, that is stable on I. The idea is that this set S will control the infinite sequences of calls on the fixed point agents $\mu_i^S\tilde{x}\tilde{e}$, for $i \in I$, on any thread of an infinite computation.

So the agent expressions of $!SCCS_\mu$ can be given using the following BNF style equation

$$\boxed{e ::= \quad x \quad | \quad a.e \quad | \quad \Sigma\tilde{e} \quad | \quad e_1 \times e_2 \quad | \quad e\lceil A \quad | \quad \mu_i^S\tilde{x}\tilde{e},}$$

The **agents of** $!SCCS_\mu$ are the agent expressions that have no free occurrences of variables. Let E_μ be the class of all agents and let K_μ be the class of the fixed point agents. We will view the class E_μ as the class E_{K_μ}, treating the elements of K_μ as the constants were treated previously. But now the recursion equation of each constant is determined by the internal structure of the constant. So if $c = \mu_i^S\tilde{x}\tilde{e} \in K_\mu$ then its defining equation $c =_{def} e_c$ is determined by letting

$$e_c = e_i[\tilde{\mu}^S\tilde{x}\tilde{e}/\tilde{x}];$$

i.e. e_c is the result of simultaneously substituting the elements of the family $\tilde{\mu}^S\tilde{x}\tilde{e} = \{\mu_i^S\tilde{x}\tilde{e}\}_{i\in I}$ for the corresponding variable in $\tilde{x}$ in the agent expression e_i.

We can now carry over the definitions of the transitions and the computations from our earlier treatment. In order to have a notion of admissible computation we will define the class T_μ of infinite sequences of elements of K_μ and then, as before, we get the notion of T_μ-admissible computation.

Let ρ be an infinite sequence of elements of K_μ. Then ρ is defined to be in T_μ if, for every family $\tilde{c} = \{c_i\}_{i\in I}$ where, for $i \in I$, $c_i = \mu_i^S\tilde{x}\tilde{e}$, if infinitely many of the terms of ρ are in $\{c_i \mid i \in I\}$ and $c_{i_0}c_{i_1}\cdots$ is the infinite subsequence of ρ of those terms then $i_0i_1\cdots$ is in S.

3.4 Semantics

We shall follow the procedure of [NWFS] and [FUP] and view labelled transition systems (LTSs) as coalgebras (A, α) for the standard functor $pow(Act \times (\ ))$ on the (superlarge) category of classes. Given such a coalgebra we write $p \xrightarrow{a} q$ when $p \in A$ and $(a, q) \in \alpha(p)$. There is a unique, up to isomorphism, final LTS and this is the final universe for $SCCS$. We want to enrich LTSs with a notion of admissible infinite computation

$$p \xrightarrow{a_0} p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots.$$

It is natural to impose the following stability condition on this notion: An infinite computation is admissible iff the computation obtained by leaving out the initial agent and first atomic action, is admissible.

So we are led to consider coalgebras (A, α) now for the functor

$$pow(Act \times (\ )) \times pow((Act \times (\ ))^\omega).$$

So α determines the two maps $\alpha^0 : A \to pow(Act \times A)$ and $\alpha^! : A \to pow((Act \times A)^\omega)$ so that for all $p \in A$

$$\alpha(p) = (\alpha^0(p), \alpha^!(p)).$$

The map α^0 gives us the LTS and the map $\alpha^!$ gives us the infinite computations. So $p \xrightarrow{a_0} p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots$ is an admissible computation if the infinite sequence $((a_0, p_0), (a_1, p_1), \ldots)$ is in $\alpha^!(p)$. If the stability condition holds then we shall call the coalgebra (A, α) a !LTS. We may now consider the full subcategory of the category of all coalgebras, consisting of the !LTSs.

Theorem: 3.6 *There is a final !LTS* $(\mathcal{P}, \eta_{\mathcal{P}})$.

Proof: We apply Theorem 2.2 of [FUP]. It is only necessary to observe that the category of !LTSs is both an image-closed and union-closed full subcategory of the category of coalgebras for a standard functor on the category of classes.

Given a syntax, with a particular choice of a class K of constants determining the class E_K of agents, with each constant having a defining equation, and with a class T of infinite sequences of constants, stable on K, we can form a !LTS $(E_K, \eta_{K,T})$, where, for $e \in E_K$,

$$\eta_{K,T} = (\eta^0_{K,T}(e), \eta^!_{K,T}(e))$$

with

$$\begin{cases} \eta^0_{K,T}(e) & = \{(a, e_0) \mid e \xrightarrow{a} e_0\} \\ \eta^!_{K,T}(e) & = \{((a_0, e_0), (a_1, e_1), \ldots) \mid e \xrightarrow{a_0} e_0 \xrightarrow{a_1} e_1 \cdots \text{ is } T\text{-admissible } \} \end{cases}$$

Now let $[[\ ]]_{K,T} : (E_K, \eta_{K,T}) \to (\mathcal{P}, \eta_{\mathcal{P}})$ be the unique !LTS map into the final !LTS $(\mathcal{P}, \eta_{\mathcal{P}})$.

Proposition: 3.7 *For* $e_1, e_2 \in E_K$

$$e_1 \sim_T e_2 \iff [[e_1]]_{K,T} = [[e_2]]_{K,T}.$$

4 A new theory of non-well-founded sets

When *Act* is a singleton set the final LTS is essentially just the final coalgebra for the standard functor *pow* and it gives the full class model of the set theory $ZFC^- + AFA$ that is the main topic of [NWFS]. We may also consider the final !LTS when *Act* is a singleton set. It turns out that this also gives a full class model of an interesting new set theory called $ZFC!AFA$ below.

For each set a let a^∞ be the set of the infinite descending $\in$-sequences from a; i.e. the set of those sequences $(a_n)_{n\in\omega}$ such that $a_0 \in a$ and $a_{n+1} \in a_n$ for all $n \in \omega$. In ZFC the Foundation Axiom FA expresses that a^∞ is always the empty set. But in theories of non-well-founded sets sets a with a^∞ non-empty are allowed. Here we introduce a variation on the theory $ZFC^- + AFA$ discussed in [NWFS]. The theory $ZFC^- + AFA$ is obtained from $ZFC^- = ZFC - FA$ by adding the Anti-foundation Axiom AFA, an axiom that is in some sense dual to the Foundation Axiom. The idea for the variant is to extend the language of set theory with a new unary postfix predicate symbol ! that we want to use to classify infinite descending $\in$-sequences into those that are deemed **admissible** and those deemed **inadmissible**. If $p = (a_n)_{n\in\omega}$ is an infinite sequence let

$p^0 = a_0$ and $p^+ = (a_{n+1})_{n\in\omega}$. We will require the following **admissibility condition** to hold for an infinite descending $\in$-sequence p:

$$p \text{ is admissible iff } p^+ \text{ is admissible}$$

The theory $ZFC!$ is obtained from the theory ZFC^- by extending the language as described, with the schemes of ZFC^- extended to the new language, and two axioms for !, the first expressing that if p is admissible then p is an infinite sequence and the second expressing the above admissibility condition. For each set a we let $a^!$ be the set of admissible elements of A^∞.

We now want to add to $ZFC!$ an axiom $!AFA$. The axiom may be stated as follows:

Every !-graph has a unique !-decoration.

This is a variant of AFA obtained by incorporating admissibility into the notions of graph and decoration of a graph used in AFA.

A !-graph, G, is a graph; i.e. a set of nodes a and a set of edges $a \to a'$ between nodes, together with a set $G^!$ of **admissible infinite paths** $a_0 \to a_1 \to \ldots$ satisfying the **admissible path condition** that an infinite path $a_0 \to a_1 \to \ldots$ is admissible iff the path $a_1 \to \ldots$, obtained by dropping the first node and edge, is admissible. A !-decoration of a !-graph G is an assignment d of a set $d(a)$ to each node a of G such that

$$d(a) = \{d(a') \mid a \to a' \text{ in } G\}$$

and

$$d(a)^! = \{(d(a_n))_{n\in\omega} \mid (a_n)_{n\in\omega} \in G^! \text{ and } a \to a_0 \text{ in } G.\}$$

4.1 Some Examples

1. Let G be the graph with one node, 0, and an edge $0 \to 0$, linking the node to itself. There is only one infinite path $0 \to 0 \to \cdots$ and there are two possible notions of admissible path for this graph so that we get two !-graphs, G_{min} and G_{max}. In G_{min} the infinite path is inadmissible and in G_{max} the infinite path is admissible. Using $!AFA$ let Ω_{min} and Ω_{max} be the two sets used to decorate the node 0 in the unique decorations of the two !-graphs. In our new set theory these are distinct solutions of the equation

   $$x = \{x\}.$$

   Note that the two sets can be distinguished because $(\Omega_{min})^! = \emptyset$ and $(\Omega_{max})^! = (\Omega_{max})_{n\in\omega}$

   This is in contrast to what happens in $ZFC^- + AFA$, where the above equation has a unique solution.

2. In general in $ZFC!AFA$ there are two possible extremes of non-well-founded set giving two subuniverses V_{min} and V_{max}, each a class model of $ZFC^- + AFA$. V_{min} is the class of those sets a where $a^! = \emptyset$ and V_{max} is the class of those sets a where $a^! = a^\infty$. So Ω_{min} is the unique solution of $x = \{x\}$ in V_{min} and Ω_{max} is the unique solution in V_{max}. Note that the sets that are in both universes are exactly the well-founded sets.

5 The Full model of $ZFC!AFA$

For each class X let X^ω be the class of all sequences $(x_n)_{n\in\omega}$ of elements $x_n \in X$. This operation on classes naturally extends to a standard functor on the (superlarge) category of classes. Using it we can define the standard functor F on the category of classes where for each class X

$$F(X) = pow(X) \times pow(X^\omega).$$

Given a coalgebra (A, α) for F, let $\alpha^0 : A \to pow(A)$ and $\alpha^! : A \to A^\omega$ be the unique maps so that for all $a \in A$

$$\alpha(a) = (\alpha^0(a), \alpha^!(a)).$$

We call the coalgebra a **!-system** if, for all $a \in A$

$$\alpha^!(a) = \{p \in A^\omega \mid p^0 \in \alpha^0(a) \text{ and } p^+ \in \alpha^!(p^0)\}.$$

A map between !-systems in the category of coalgebras for F will be called a **!-system map**.

Theorem: 5.1 *There is a final !-system. Moreover it determines a full class model of $ZFC!AFA$.*

Proof: The category of !-systems is a full subcategory of the category of coalgebras for the standard functor F. We can apply Theorem 2.2 of [FUP] to get that there is a final !-system because the full subcategory is image closed and union closed, as defined in [FUP].

Now let (A, α) be a final !-system. I show that (A, α^0) is full; i.e. that if $x \in pow(A)$ then there is a unique $c \in A$ such that $\alpha^0(c) = x$.

Let A_0 be the smallest subset of A such that $x \subseteq A_0$ and for all $y \in A_0$ $\alpha^0(y) \subseteq A_0$. Also choose $* \notin A$ and let $B = A_0 \cup \{*\}$. We can now define $\beta : B \to pow(B) \times pow(B^\omega)$ by

$$\begin{cases} \beta(a) & = \alpha(a) \text{ for } a \in A_0 \\ \beta(*) & = (x, \{y \frown \sigma \mid y \in x \ \& \ \alpha^!(y)\}) \end{cases}$$

Claim: (B, β) is a !-system.

Let $\pi : (B, \beta) \to (A, \alpha)$ be the unique !-system map.

Claim: $\pi(a) = a$ for all $a \in A_0$.

Let $c = \pi(*)$. Then

$$\begin{aligned} \alpha^0(c) & = \{\pi(a) \mid a \in \beta^0(*)\} \\ & = \{\pi(a) \mid a \in x\} \\ & = x \end{aligned}$$

Finally, to show the uniqueness of c let $c' \in A$ such that $\alpha^0(c') = x$. The we may define $\pi' : B \to A$ by

$$\begin{cases} \pi'(a) & = a \text{ for all } a \in A_0 \\ \pi'(*) & = c' \end{cases}$$

Claim: π' is a !-system map $(B, \beta) \to (A, \alpha)$.

Hence $\pi' = \pi$ so that $c' = \pi'(*) = \pi(*) = c$ and the proof that (A, α^0) is a full system is finished. It follows that a full class model of ZFC^- is now determined in the usual way.

6 !-bisimulations

Let (A, α) be a !-system. A relation $\sim$ on A is a **!-bisimulation** if whenever $a \sim b$

$$\forall a' \in \alpha^0(a)\ \exists b' \in \alpha^0(b)\ \ a' \sim b',$$

$$\forall b' \in \alpha^0(b)\ \exists a' \in \alpha^0(a)\ \ a' \sim b',$$

$$\forall p \in \alpha^!(a)\ \exists q \in \alpha^!(b)\ \ p \sim^\infty q,$$

and

$$\forall q \in \alpha^!(b)\ \exists p \in \alpha^!(a)\ \ p \sim^\infty q,$$

where if $p = (a_n)_{n\in\omega}$ and $q = (b_n)_{n\in\omega}$ then

$$p \sim^\infty q \ \Leftrightarrow\ \forall n\ a_n \sim b_n.$$

Theorem: 6.1 *There is a largest !-bisimulation, $\sim_A$, on (A, α).*

(A, α) is **!-extensional** if $a \sim_A b$ implies that $a = b$.

7 References

NWFS Peter Aczel, *Non-Well-Founded Sets*, CSLI Lecture Notes No. 14, Stanford, 1988.

FUP Peter Aczel, *Final Universes of Processes*, in Springer Lecture Notes in Computer Science No. 802, 1994.

Hart1 Chrysafis Hartonas, *Semantics for Finite Delay*, To appear in Theoretical Computer Science.

Hart2 Chrysafis Hartonas, *A Calculus for Finite Delay and Fairness*, Draft paper, March, 1996.

Henn1 Matthew Hennessy, *Axiomatising Finite Delay Operators*, Technical Report CSR-124-82, University of Edinburgh ,Department of Computer Science, August 1982.

Miln Robin Milner, *A Finite Delay Operator in Synchronous CCS*, Technical Report CSR-116-82, University of Edinburgh, Department of Computer Science, May 1982.