Modal Logic and Equality for Process Algebra

William P.R.Mitchell
Department of Computer Science
University of Manchester
Manchester M13 9PL
bill@cs.man.ac.uk

Febuary 22, 1994

Abstract

This paper provides a denotational semantics for the saturation method of [**2**] in the form of a modal logic. In particular the semantics describes congruence for CCS processes in the form of a modal logic whose axioms are a modal representation of the τ-laws. The main idea of the paper is to represent bisimulation equivalences of CCS in terms of a simple modal logic $\mathbf{K}^{\oplus}$. This logic is a variant of the classical modal logic $\mathbf{K}$ introducing a plus operator to mirror the choice operator of CCS. Weak bisimulation equivalence and process equality can be characterised in terms of extra axioms added to $\mathbf{K}^{\oplus}$. In the case of equality the axioms are a modal representation of the τ-laws. A CCS process maps to a model of the logic; two processes are equivalent if they map to the same model.

1 Introduction

Milner introduced the world to process algebras in his book [**8**], in which a process is represented algebraically as a term in a calculus. The book also gives an account of various equivalence relations between calculus terms: strong bisimulation equivalence $\sim$, weak bisimulation equivalence $\approx$ and equality $=_\tau$. Since then much work has been done on finding a deeper understanding of these equivalences by explaining CCS in terms of various logics, particularly multi-modal logics.

In an algebraic setting, when considering finite processes, equality can be axiomatised by three simple equations known as the τ-laws:

$$\begin{aligned} a \cdot x &=_\tau a \cdot \tau \cdot x \\ \tau \cdot x &=_\tau \tau \cdot x + x \\ a \cdot (x + \tau \cdot y) &=_\tau a \cdot (x + \tau \cdot y) + a \cdot y \end{aligned}$$

These do not axiomatise equality for infinite processes, however the saturation method [**2**] gives a geometrical interpretation to these equations which does characterise $=_\tau$ for all processes.

A process determines a transition system which can be regarded as a directed edge labeled graph. The saturation method regards the τ-laws as a set of graph rewrite rules. For any graph G the rules generate a canonical normal form G^* known as the saturated graph for G. The graph G^* is generated by exhaustively rewriting G. Two processes are equal in the sense of [**8**] if and only if they have the same saturated graph. A similar method can be used to define weak equivalence.

Algebraically the τ-laws seem somewhat arbitrary, but in the saturation method they become quite natural. As graph rewrite rules they become the following; where τ represents the silent action, u, v and w are nodes, and $\xrightarrow{a}$ is an a labeled edge:

$$\begin{array}{l}
\text{Given } u \xrightarrow{a} v \xrightarrow{\tau} w \text{ add } u \xrightarrow{a} w \\
\text{Given } u \xrightarrow{\tau} v \xrightarrow{a} w \text{ add } u \xrightarrow{a} w \\
\text{Given } u \xrightarrow{a} v \text{ add } v \xrightarrow{\tau} v
\end{array}$$

The first and second rules state that any τ-transition can be bypassed, and the third says that after any transition it is possible to idle for an arbitrary amount of time before the next action.

The saturation method gives a simple geometric explanation of equality which at the same time gives an intuitively simple understanding of the τ-laws.

The main goal of this paper is to present the geometrical insight of the saturation method in a modal logic setting. CCS is given a denotational semantics as a modal logic which also contains the saturation method. Moreover in the case of equality this semantics equates CCS with a modal logic axiomatised by a modal version of the τ-laws. Work on a similar line to this, but in a category theory context, can be found in a recent paper [**1**], which includes a representation of the saturation method as a final universe of a category characterising CCS.

The other goal of this paper is to make the underlying characterising logic for CCS as conceptually simple as possible. A process performs actions, the chronological ordering of these actions is somehow determined by the process. In CCS the calculus term encodes this ordering and what possible actions there are.

The simplest way to represent such a chronological ordering is as a set of worlds which are connected by a flow of time. Each world represents a moment of time where some action occurs. This is the definition of a Kripke model. The simplest modal operators over such models are $\Box$ always, and $\Diamond$ sometimes. The logic $\mathbf{K}$ consists of the set of Kripke models and formulae made from the standard propositional logic connectives together with $\Box$ and $\Diamond$, see [**7**].

The idea for this paper is to start with a generic modal logic $\mathbf{K}^{\oplus}$ which is $\mathbf{K}$ with an additional modal operator $\oplus$. Extensions to $\mathbf{K}^{\oplus}$ are defined by adding simple axioms which are the assumptions for the silent action τ. Two extensions of the logic are defined in this way $\mathcal{M}_{el}$ and $\mathcal{M}_{ev}$. Logic $\mathcal{M}_{ev}$ will characterise $\approx$ and $\mathcal{M}_{el}$ will characterise $=_{\tau}$. In the case of $\mathcal{M}_{el}$ the axioms represent a modal version of the τ-laws, at the same time the axioms represent a modal version of the saturation rewrite rules (see definition 3.1 and proposition 3.2).

A calculus term of CCS can be mapped to a model in each of these logics in a natural way (see definition 3.4). Proposition 4.2 proves that the saturation process is equivalent to this mapping. The mapping comes from the idea of one process simulating another. Theorem 3.5 proves that two processes are equal (respectively weakly equivalent) if and only if they map to the same model in $\mathcal{M}_{el}$ (respectively $\mathcal{M}_{ev}$).

2 Related Work, Multi-modal Logics

Ever since process algebras first appeared it was obvious that $\sim$ and $\approx$ had some deeper meaning which could be explained in terms of multi-modal logic. The first such work appeared in [**5**]. For a reasonably clear account of some of the other logics the reader is referred to [**11**]. This short section provides a thumb nail sketch of multi-modal logics and their differences with $\mathcal{M}_{el}$ and $\mathcal{M}_{ev}$.

A model of multi-modal logic consists of a graph $G = (U, E)$, with edges E and nodes U. Each edge is labeled by a letter from a set A, each node u is labelled by a truth assignment $V_u : P \longrightarrow \mathbb{B}$, which is a map from a set of propositional variables to the booleans. Each model also has some of its nodes labeled as start nodes.

For two models M, N let $M \xrightarrow{a} N$ denote that N is a maximal connected component of M generated by a node which is connected to a start node by an edge labeled with a. That is there is a start node s, an edge $s \xrightarrow{a} v$, v is a node of N, and a node is in N if and only if it is a descendent of v.

A transition system generated by a CCS process term is a graph with edges labeled by action names A. This can trivially be made into a model by giving every node the valuation which makes every propositional variable true. Let $M(P)$ be the model given by a process P.

The simplest form of modalities of multi-modal logic are $[a]$ and $\langle a \rangle$. For a model M and formula f the semantics are given by

1. $M \models [a]f$ iff $\forall M \xrightarrow{a} N$, $N \models f$
2. $M \models \langle a \rangle f$ iff $\exists M \xrightarrow{a} N$, $N \models f$

Note that there is no operator $\Box$. This would have the semantics $\Box f = \forall a \in A\, [a]f$.

With these simple operators it is possible to characterise $\sim$ for finite processes when the set of actions A is also finite. Let θ : CCS $\longrightarrow$ multi-modal logic, be defined by

$$\begin{aligned} \theta(\sum_{x \in X} x \cdot P_x) &= \textstyle\bigwedge_{x \in X} [x]\theta(P_x) \wedge \bigwedge_{y \in A - X} [y]F \\ \theta(0) &= \textstyle\bigwedge_{y \in A} [y]F \end{aligned}$$

where 0 is the empty process, and F is the boolean false. Two processes P and Q are strongly bisimulation equivalent if and only if for every multi-modal logic model M, $M \models \theta(P) \Leftrightarrow M \models \theta(Q)$. It is also the case that $P \sim Q$ if and only if for every modal formula f, $M(P) \models f \Leftrightarrow M(Q) \models f$.

If the set of actions is infinite one possible way to extend the logic is to allow infinite conjunctions.

An alternative to this very basic multi-modal logic is the modal like logic [**3**]. The full details are not given here, only a crude approximation. A model of [**3**] is a transition tree, that is a tree with labeled edges. The modal operators are very like $\Box$ and $\Diamond$, but there is also an operator $+$. Given two trees t_1 and t_2, $t_1 + t_2$ is formed by identifying together the two roots of t_1 and t_2. The semantics are summed up as follows

1. $M \models \Box f$ iff $\forall\ M \xrightarrow{a} N$, $N \models f$
2. $M \models \Diamond f$ iff $\exists\ M \xrightarrow{a} N$, $N \models f$
3. $t \models f + g$ iff $t = t_1 + t_2$ and $t_1 \models f$ and $t_2 \models g$
4. For an action $a \in A$, $M \models a$ iff $\exists\ M \xrightarrow{a} N$.

Using these simple operators it is much easier to characterise a process as a logical formula. The map θ can be redefined so that

$$\theta(\sum_{x \in X} x \cdot P_x) = \sum_{x \in X} \Box(x \wedge \theta(P_x))$$

The map θ captures $\sim$ just as before.

Defining $\sim$ in terms of a multi-modal logic is not particularly interesting, however the same approach also works for $\approx$, but with a different set of modal operators. Let $\triangle f$ be a fix-point formula defined by $\triangle f = [\tau]\triangle f \vee f$, and let $\lfloor a \rceil f = \triangle [a] f$. Redefine θ again, this time to characterise $\approx$.

$$\begin{aligned} \theta(\sum_{x \in X} x \cdot P_x) &= \textstyle\bigwedge_{x \in X} \lfloor x \rceil \theta(P_x) \wedge \bigwedge_{y \in A - X} \lfloor y \rceil F \\ \theta(0) &= \textstyle\bigwedge_{y \in A} \lfloor y \rceil F \end{aligned}$$

For processes P and Q, $P \approx Q$ iff for every multi-modal logic model M, $M \models \theta(P) \Leftrightarrow M \models \theta(Q)$.

The paper [**3**] also provides a modal like logic which characterises $\approx$. As with the multi-modal logics this is acheived by extending the simple form of the logic with fix-point formulae. Analogous operators are defined to those of the above multi-modal logic; $\triangle f = f \vee \Box(\tau \wedge f)$ and $\lfloor a \rceil f = \triangle \Box (a \wedge f)$. The map θ characterising $\approx$ for this modal setting is defined in exactly the same way as for the multi-modal setting. The $\triangle$ operator builds into the logic the inherant structure of the $\approx$ equivalence. This modal logic can be further extended to characterise $=_\tau$, as with $\approx$ the structure of $=_\tau$ is built into the modalities.

In order to get around most of the problems with infinite processes the modal and multi-modal logics can be further extended by allowing arbitrary fix-point operators.

All of the above approches take their set of models to be the universe of transition graphs, or the universe of transition trees. The understanding they provide for observational equivalences is encoded by the modal operators of the logic. The model $M(P)$ or the formula $\theta(P)$ does not contain any information not present in the syntactic structure of P, only the interpretation of the modalities provides any deeper understanding. There is also no immedialty apparent connection between the τ-laws or saturation and any of the above logics.

The logic $\mathbf{K}^{\oplus}$ has as models the set of (possibly infinite) trees, where each node is labeled by a valuation function $A \longrightarrow \mathbb{B}$, A being a set of propositional variables and also the set of action names. The modalities of $\mathbf{K}^{\oplus}$ are $\Box$, $\Diamond$ and $\oplus$. The modality $\oplus$ is related to the $+$ of [**3**], it is defined precisly in the next section. The logics $\mathcal{M}_{ev}$ and $\mathcal{M}_{el}$ are each defined by a particular subset of the universe of models. Thus not every model qualifies as part of $\mathcal{M}_{el}$ or $\mathcal{M}_{ev}$. A model is accepted by one of these logics if each of the defining axioms for the logic is true everywhere in the model.

In the next section a cetain partial ordering is defined for models of $\mathbf{K}^{\oplus}$. Each process P, whether finite or infinite, can be regarded as a node labeled tree $M(P)$, and thus as a model of $\mathbf{K}^{\oplus}$. A process P can be identified with the minimal model $\mathcal{M}_{ev}(P)$ (or $\mathcal{M}_{el}(P)$) of $\mathcal{M}_{ev}$ (or $\mathcal{M}_{el}$) which contains $M(P)$. It is proved later that this model corresponds to the result of saturating $M(P)$.

For processes P and Q, $P \approx Q$ if and only if for every formula f of $\mathbf{K}^{\oplus}$,

$$\mathcal{M}_{ev}(P) \models f \Leftrightarrow \mathcal{M}_{ev}(Q) \models f$$

Also $P =_{\tau} Q$ if and only if for every formula f of $\mathbf{K}^{\oplus}$,

$$\mathcal{M}_{el}(P) \models f \Leftrightarrow \mathcal{M}_{el}(Q) \models f$$

In fact it turns out that $P \approx Q$ iff $\mathcal{M}_{ev}(P) \sim \mathcal{M}_{ev}(Q)$, and that $P =_{\tau} Q$ iff $\mathcal{M}_{el}(P) \sim \mathcal{M}_{el}(Q)$ where $\sim$ is a relation on models induced by strong bisimulation equivalence.

The approach of this paper is to characterise $\approx$ and $=_{\tau}$ in terms of logics $\mathcal{M}_{ev}$ and $\mathcal{M}_{el}$; each logic defined as a certain set of models which satisfy particular axioms. This is different from the earlier logic characterisations where any model is acceptable, and the logic is defined as a set of particular formulae, formulae constructed out of particular modalities which encode the observational equivalence.

The axioms of $\mathcal{M}_{el}$ and $\mathcal{M}_{ev}$ are intimately related to the τ-laws and to saturation, there is no such immediate connection with the earlier logics.

The complete structure which a process P encodes with repect to an observational equivalence is encoded within the model $\mathcal{M}_{ev}(P)$ or $\mathcal{M}_{el}(P)$. This is different from earlier logics where the structure of P is encoded by the semantics of the modalities together with the model identified with P, this model containing just the syntactic structure of P.

Their is a certain duality between the ealier characterisations of process equivalence and the $\mathcal{M}_{el}$, $\mathcal{M}_{ev}$ characterisations. With the earlier logics, the logic encodes a lot of information about the particular equivalence, and the process is represented in a straighforward manner. With the $\mathcal{M}_{el}$, $\mathcal{M}_{ev}$ characterisation the logic encodes little of the equivalence (i.e just what is in the axioms), and the the process is represented in a way which encodes almost all the structure of the equivalence.

The earlier logics encode the properties of observational equivalence within their modalities. The approach here is to encode the τ-laws (saturation rewrite rules) as a set of axioms which are part of an elementary logic with a minimal number of modalities.

3 A Modal Logic with Choice

This section defines the generic modal like logic $\mathbf{K}^{\oplus}$ which will be extended to characterise $\approx$ and $=_\tau$. The modal axioms for $\approx$ and $=_\tau$ are defined and the characterisation of a process as a model for these logics is described.

Let A be an infinite set of action names, which will also be used as a set of propositional variables. Included in this set is the distinguished variable τ. Define $\mathcal{M}$ to be the set of trees with nodes labeled by subsets of A. Let $V(u) \subseteq A$ be the set of variables which label node u. Regard $a \in A$ as true at node u if $a \in V(u)$ and false at u if $a \notin V(u)$. Trees of $\mathcal{M}$ must be finitely branching, but branches may be infinite in length. A tree M_1 is a child of M if M_1 is a subtree of M and the parent node to the root of M_1 is the root of M. The tree 0 has root node labeled by $\emptyset$ and no children.

For two trees M_1, $M_2 \in \mathcal{M}$ let $M_1 \oplus M_2$ be the tree formed by 'gluing' the two trees together at the root, the label for the root of $M_1 \oplus M_2$ is the union of the labels for the roots of M_1 and M_2. If M_1 has root label A_1 with children T_i for $1 \leq i \leq n$ and M_2 has root label A_2 and children T_i for $n+1 \leq i \leq n+m$, then $M_1 \oplus M_2$ has root label $A_1 \cup A_2$ and children T_i for $1 \leq i \leq m+n$.

Define a congruence relation $\sim$ over $\mathcal{M}$ by the following axioms (they are precisely the axioms which define strong bisimulation equivalence for processes).

commutative	$M_1 \oplus M_2 \sim M_2 \oplus M_1$
associative	$(M_1 \oplus M_2) \oplus M_3 \sim M_1 \oplus (M_2 \oplus M_3)$
identity	$M \oplus 0 \sim M$
idempotent	$M \oplus M \sim M$

These axioms simply specify that the order of subtrees is unimportant and repetition is redundant.

For a tree M and node u, M/u is the subtree of M starting from node u. For u, v nodes of M let $u \longrightarrow v \in M$ denote that there is an edge from u to v in M. For M, $M' \in \mathcal{M}$ with λ the root of M, we write $M \longrightarrow M'$ to denote that for some $\lambda \longrightarrow u \in M$, $M/u = M'$. That is M' is a child of M.

Let $\Box$, $\Diamond$ and $\oplus$ be modal operators and $\wedge$, $\vee$ and $\neg$ be the usual logical connectives. Operator $\oplus$ is an infix operator. Let $\mathbf{K}^{\oplus}$ be the set of formulae constructed by these operators. They are semantically defined over $\mathcal{M}$ as follows.

For $M \in \mathcal{M}$, f, $g \in \mathbf{K}^{\oplus}$ and u a node of M the satisfaction relation $\models$ is defined by:

- $(u, M) \models f \oplus g$ if there are M_1, $M_2 \in \mathcal{M}$ with λ_i the root of M_i such that
$$(\lambda_1, M_1) \models f,\ (\lambda_2, M_2) \models g \ \text{ and } \ M/u \sim M_1 \oplus M_2$$
- $(u, M) \models \Box f$ if for every $u \longrightarrow v \in M$, $(v, M) \models f$
- $(u, M) \models \Diamond f$ if for some $u \longrightarrow v \in M$, $(v, M) \models f$

The definition of $\models$ is as usual for the propositional connectives. For a propositional variable p define $(u, M) \models p$ if $p \in V(u)$. For $M \in \mathcal{M}$ define $M \models f$ if and only if for every node u of M, $(u, M) \models f$. The anchored version of satisfaction is denoted $\models_0$. Define $M \models_0 f$ if $(\lambda, M) \models f$ where λ is the root node of M. For a set of models $S \subseteq \mathcal{M}$ define $S \models f$ if and only if for every $M \in S$, $M \models f$. This logic, which we shall refer to as $\mathbf{K}^{\oplus}$ is like the classical modal logic $\mathbf{K}$ (see [**7**] for the definition) with one extra operator $\oplus$.

When writing out formulae $\oplus$ is given priority over $\wedge$ and $\vee$, for example $x \oplus y \vee x \oplus z$ reads as $(x \oplus y) \vee (x \oplus z)$. Always $\Box$ and sometime $\Diamond$ have highest priority, for example $\Box x \oplus \Diamond y$ reads as $(\Box x) \oplus (\Diamond y)$.

As a convenient shorthand write $\Diamond_a f$ as an abbreviation for $\Diamond(a \wedge f)$ and $\Box_a f$ as an abbreviation for $\Box(a \wedge f)$. This shorthand requires some caution when simplifying formulae. For example $\neg\Diamond_a f \not\equiv \Box_a \neg f$. It may also cause some confusion as $\Box_a$ bears a passing resemblance to $[a]$ which is a multi-modal operator. It is important to remember that $\Box_a$ is syntactic sugar and not a modal operator in its own right.

The following equivalences illustrate how $\oplus$ mixes with the other connectives.

$$\begin{array}{lll}
\mathcal{M} \models & x \oplus (y \vee z) & \leftrightarrow & (x \oplus y \vee x \oplus z) \\
\mathcal{M} \models & \Box x \oplus \Diamond y & \leftrightarrow & \Diamond x \wedge \Diamond y \\
\mathcal{M} \models & \Diamond x \oplus \Diamond y & \leftrightarrow & \Diamond x \wedge \Diamond y \\
\mathcal{M} \models & \Box x \oplus \Box y & \leftrightarrow & \Box (x \vee y) \\
\mathcal{M} \models & \Box_a x \oplus \Box_a y & \leftrightarrow & \Box_a (x \vee y)
\end{array}$$

One minor drawback of introducing $\oplus$ from a purely logical perspective is that there appears to be no general way of simplifying $\neg(x \oplus y)$. This means there is no prenex normal form in the usual sense. As a result duality of $\Box$ and $\Diamond$ is not as comprehensive as usual in $\mathbf{K}^{\oplus}$. For example from the fourth equivalence above duality would suggest that there is a formula for distributing $\Diamond$ over a conjunct using $\oplus$, but in fact this is not the case.

Definition 3.1 Define $\mathcal{M}_{ev} \subseteq \mathcal{M}$ to be the set of models such that $\forall a, b \in A$, $\forall x \in \mathbf{K}^{\oplus}$

1. $\mathcal{M}_{ev} \models \Diamond_a \Diamond_\tau x \leftrightarrow \Diamond_a x$
2. $\mathcal{M}_{ev} \models \Diamond_\tau x \leftrightarrow (\Diamond_\tau x \oplus x)$
3. $\mathcal{M}_{ev} \models \Diamond_a(\Diamond_\tau x \oplus y) \leftrightarrow (\Diamond_a(\Diamond_\tau x \oplus y) \oplus \Diamond_a x)$
4. $\forall a \neq b$, $\mathcal{M}_{ev} \models a \rightarrow \neg b$
5. $\mathcal{M}_{ev} \models x \rightarrow \Diamond_\tau x$

To define $\mathcal{M}_{el}$ modify the definition of $\mathcal{M}_{ev}$ by replacing $\mathcal{M}_{ev}$ with $\mathcal{M}_{el}$ in axioms 1, 2 and 3 and discarding axiom 5.

Axioms 1, 2, 3 above are clearly a modal form of the τ-laws. They will be referred to as the modal τ-laws. The following theorem gives a much simpler and rather more elegant set of axioms for $\mathcal{M}_{el}$. These alternative axioms are an indirect translation of the saturation rewrite rules into modal form.

Proposition 3.2 An equivalent set of axioms to those which define $\mathcal{M}_{el}$ are axiom 4 above plus,

$$\begin{array}{ll} \Diamond_a \Diamond_\tau x \leftrightarrow \Diamond_a x & (f_1) \\ \Diamond_\tau \Diamond_a x \rightarrow \Diamond_a x & (f_2) \end{array}$$

Proof Let τ_1, τ_2, τ_3 denote the three modal τ-laws. Note $\tau_1 = f_1$. It is a fact that $\mathbf{K}^{\oplus} \models (f \rightarrow g) \rightarrow (f \rightarrow f \oplus g)$, also $\mathbf{K}^{\oplus} \models \Diamond f \oplus g \rightarrow \Diamond f$. The following uses these two facts to prove that τ_3 is a logical consequence in $\mathbf{K}^{\oplus}$ of τ_1.

$$\begin{array}{ll} \Diamond_a(\Diamond_\tau x \oplus y) \rightarrow \Diamond_a \Diamond_\tau x & (\text{since } \Diamond_\tau x \oplus y \rightarrow \Diamond_\tau x) \\ \Diamond_a \Diamond_\tau x \rightarrow \Diamond_a x & (\text{by axiom 1}) \\ \therefore \Diamond_a(\Diamond_\tau x \oplus y) \rightarrow \Diamond_a x & \\ \therefore \Diamond_a(\Diamond_\tau x \oplus y) \rightarrow \Diamond_a(\Diamond_\tau x \oplus y) \oplus \Diamond_a x & \end{array}$$

Clearly $\Diamond_a(\Diamond_\tau x \oplus y) \oplus \Diamond_a x \rightarrow \Diamond_a(\Diamond_\tau x \oplus y)$, hence the axiom τ_3 is redundant as it is a logical consequence in $\mathbf{K}^{\oplus}$ of τ_1. To complete the proof it is necessary to show that for every model M, and every node u of M,

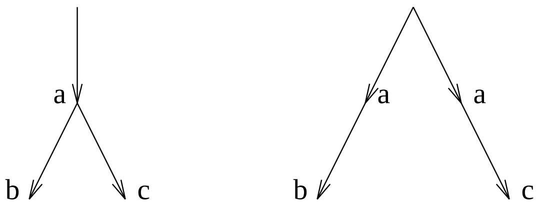
$$\forall x,\ ((u, M) \models \tau_2) \Leftrightarrow \forall x,\ ((u, M) \models f_2) \tag{1}$$

To prove this we first describe how to characterise a tree by formulae of $\mathbf{K}^{\oplus}$. Let $\Box\!\Box f$ be shorthand for $\Box f \wedge \Diamond$ True . The formula $\Box\!\Box f$ states that in every future world f holds, and that future worlds do exist. Modality $\Box\!\Box$ is pronounced strong always. Let $\Box\!\Box_a f$ denote $\Box\!\Box(a \wedge f)$, and $\Box\!\Box_a$ on its own denote $\Box\!\Box a$. Let $\sum_{i=1}^{n} f_i$ denote $f_1 \oplus f_2 \oplus \cdots f_n$.

Every finite tree T can be characterised by a single formula $\psi(T)$ of $\mathbf{K}^{\oplus}$. Let T be a finite tree with root label $S \subseteq A$ and children T_i for $1 \leq i \leq n$. Let

$$\psi(T) = \bigwedge_{a \in S} a \wedge \sum_{i=1}^{n} \Box\!\Box \psi(T_i)$$

and let $\psi(0)$ =True. For example the two trees below have characterising formulae $\Box\!\Box_a(\Box\!\Box_b \oplus \Box\!\Box_c)$ and $\Box\!\Box_a \Box\!\Box_b \oplus \Box\!\Box_a \Box\!\Box_c$ respectively.



For two finite trees T_1 and T_2 it is the case that $T_1 \sim T_2$ if and only if $\mathcal{M} \models_0 \psi(T_1) \leftrightarrow \psi(T_2)$. In fact if $M \models_0 \psi(T)$ then $M \sim T$. Define tree T to be a d-approximation of S if by deleting subtrees of S which occur at depth $d+1$ we construct T. Tree T is an approximation to S if for some d, T is a d-approximation of S. The set of formulae $\psi(S)$ which characterise an infinite tree S are the formulae $\psi(T)$ which characterise the finite approximations T of S.

If $\mathbf{K}^{\oplus}$ were generalised to allow fix point formulae, infinite observation trees would have a single defining formula. However, one of the aims of the paper is to keep $\mathbf{K}^{\oplus}$ as simple as possible, so fix point formulae are left out, even though this makes the proof slightly more cumbersome. For an infinite tree S, let $M \models_0 \psi(S)$ denote $\forall f \in \psi(S)$, $M \models_0 f$. For any trees M and M', $M' \models_0 \psi(M)$ if and only if $M \sim M'$.

For a tree M and a node u let M/u denote the subtree of M whose root is the node u.

Proof of 1. In $\mathbf{K}^{\oplus}$,

$$f_2 \leftrightarrow (\Diamond_\tau \Diamond_a x \leftrightarrow \Diamond_\tau \Diamond_a x \oplus \Diamond_a x)$$

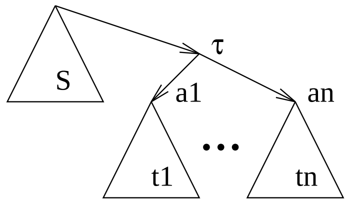
Let $f_3 = \Diamond_\tau \Diamond_a x \leftrightarrow \Diamond_\tau \Diamond_a x \oplus \Diamond_a x$. Therefore to prove 1 it is enough to prove

$$\forall x\,((u, M) \models \tau_2) \Leftrightarrow \forall x\,((u, M) \models f_3) \tag{2}$$

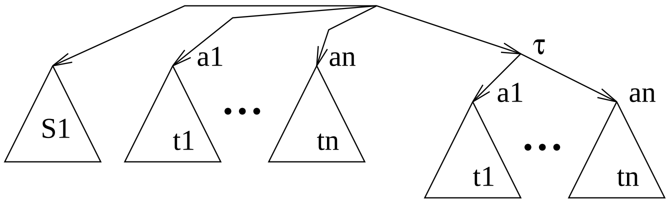
Formula f_3 is clearly a special case of τ_2, so the proof is only required for the $\Leftarrow$ direction.

Suppose $\forall x$, $(u, M) \models f_3$. Formula $\Diamond_\tau x \oplus x \to \Diamond_\tau x$ is valid so the proof is complete if we prove $\forall x$, $(u, M) \models \Diamond_\tau x \to \Diamond_\tau x \oplus x$.

This statement is vacuous if $(u, M) \not\models \Diamond_\tau$, so assume $(u, M) \models \Diamond_\tau$. In this case M/u has the form



It follows that $(u, M) \models \Diamond_\tau \Diamond_{a_i} \psi(t_i)$ for $1 \leq i \leq n$, and so $(u, M) \models \Diamond_\tau \Diamond_{a_i} \psi(t_i) \oplus \Diamond_{a_i} \psi(t_i)$ for $1 \leq i \leq n$. This can only be if M/u has the form



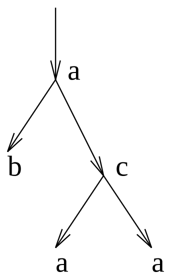
From this it follows that if $(u, M) \models \Diamond_\tau x$ then $(u, M) \models \Diamond_\tau x \oplus x$, as required. This completes the proof.

We regard $\mathcal{M}_{ev}$ and $\mathcal{M}_{el}$ as defining modal logics. The logic defined by $\mathcal{M}_{ev}$ (respectively $\mathcal{M}_{el}$) has as valid formulae those $f \in \mathcal{M}$ such that $\mathcal{M}_{ev} \models f$ (respectively $\mathcal{M}_{el} \models f$). We use $\mathcal{M}_{ev}$ (respectively $\mathcal{M}_{el}$) as the name for the formulae of the logic and for the set of models which define the logic. The logic $\mathcal{M}_{ev}$ will be used to characterise $\approx$ (*ev* is to be read as an abbreviation of *equivalence*), and $\mathcal{M}_{el}$ will be used to characterise $=_\tau$ (*el* is to be read as an abbreviation for *equal*).

Leaving the proofs aside for the moment it is interesting to explain how these logics characterise $\approx$ and $=_\tau$.

Definition 3.3 Let P be a process, let $P \xrightarrow{a_i} P_i$ for $1 \leq i \leq n$ be the complete set of transitions for P. Define $M(P) \in \mathcal{M}$ to have root node labeled by $\emptyset$, and to have n children which are the trees $M'(P_i)$ for $1 \leq i \leq n$, where the tree $M'(P_i)$ is constructed from $M(P_i)$ by changing the root label to $\{a_i\}$.

For example the tree for $a \cdot (b \cdot 0 + c \cdot (a \cdot 0 + a \cdot 0))$ is given by

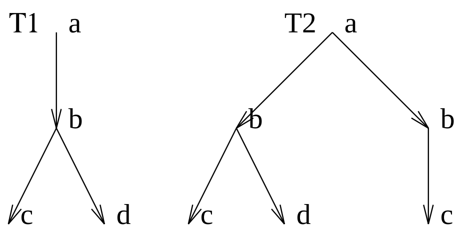


The root node is labeled with $\emptyset$ but the convention is to leave this out of the picture. The tree $M(P)$ is the observation tree of P. Notice that $M(P)$ is almost identical to the derivation tree [**8**] $T(P)$ (this is the transition system for P unwound into a tree). Any derivation tree converts into an observation tree by 'sliding' each edge label along to the node which the edge points to.

For M_1, $M_2 \in \mathcal{M}$, let M_i have root labeled by S_i and have children T_j^i for $1 \leq j \leq n_i$. Define $M_1 \subseteq M_2$ if

1. $S_1 \subseteq S_2$
2. there is an injection $f : \{1, \ldots, n_1\} \hookrightarrow \{1, \ldots, n_2\}$ such that $T_j^1 \subseteq T_{f(j)}^2$ for each $1 \leq j \leq n_1$.

The $\subseteq$ relation is very close to the subtree relation. Roughly $M_1 \subseteq M_2$ if by some suitable permutation of all the subtrees in M_1, a new tree is constructed which is an exact subtree of M_2 occurring at the root. For example with the trees depicted below, $T1 \subseteq T2$ but $T2 \not\subseteq T1$.



The relation $\subseteq$ is a partial ordering of $\mathcal{M}$. In the following definition the min of a set is defined with respect to $\subseteq$. The relation $\subseteq$ is closely related to the idea of simulation. If $M_1 \subseteq M_2$, then M_2 represents a tree which can simulate all the actions of M_1.

Definition 3.4 Define

$$\mathcal{M}_{ev}(P) = \min \{M \in \mathcal{M}_{ev} \mid M(P) \subseteq M\}$$

$$\mathcal{M}_{el}(P) = \min \{M \in \mathcal{M}_{el} \mid M(P) \subseteq M\}$$

The model $\mathcal{M}_{ev}(P)$ (respectively $\mathcal{M}_{el}(P)$) is the smallest model containing $M(P)$ which satisfies the defining axioms for $\mathcal{M}_{ev}$ (respectively $\mathcal{M}_{el}$). It will be shown later that this model is constructed via the saturation process given by [**2**]. Notice that the definition is well defined up to $\sim$.

Theorem 3.5 Let P, Q be processes in CCS then

1. $P \approx Q$ if and only if $\mathcal{M}_{ev}(P) \sim \mathcal{M}_{ev}(Q)$.
2. $P =_\tau Q$ if and only if $\mathcal{M}_{el}(P) \sim \mathcal{M}_{el}(Q)$.

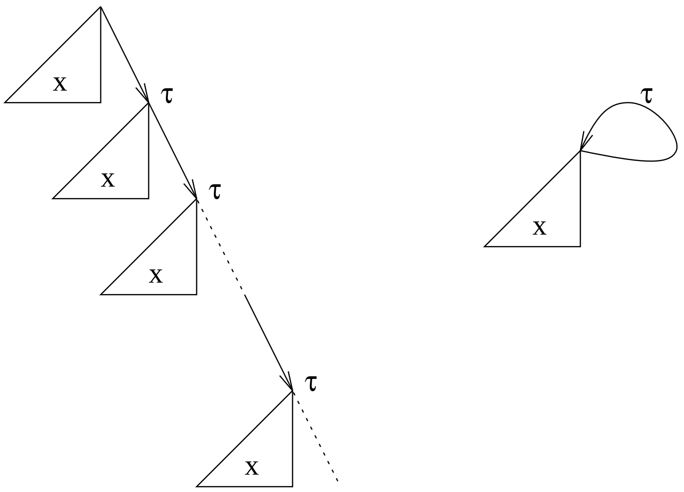
This theorem is proved in the next section. Theorem 3.5 proves the two observational equivalences can be characterised in terms of mappings into modal logics. These logics are appealing in that they are simple extensions to a very basic modal logic $\mathbf{K}^{\oplus}$ by axioms which are natural for CCS. The mapping is itself very simple.

4 Saturation of Models

This section provides a constructive definition for $\mathcal{M}_{ev}(P)$ and $\mathcal{M}_{el}(P)$ in terms of the saturation process of [**2**] and proves theorem 3.5. The results of [**2**] apply to transition systems which can be described in terms of directed graphs with edge labels. Such a graph is equivalent to the (possibly infinite) tree which is obtained by unwinding the loops of the graph. As mentioned earlier an edge labeled tree converts to a node labeled tree. Therefore the transition systems of [**2**] can be regarded as models in $\mathcal{M}$. The saturation process can then be directly translated into the modal framework here.

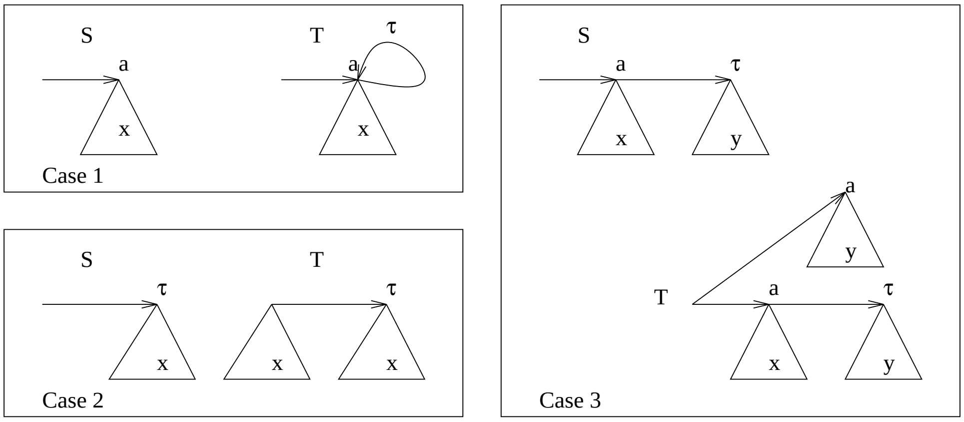
For trees S and M write $S \leq M$ if $M \sim S \oplus S'$ for some S'. For a tree M where $S \leq M/u$ let $M[u, T/S]$ be constructed out of M by replacing the occurrence of S at node u with the tree T. If $M/u \sim M' \oplus S$ then $M[u, T/S]$ is the result of replacing M/u with $M' \oplus T$ in M, thus $M[u, T/S]/u \sim M' \oplus T$.

As a useful graphical abbreviation we use the picture below on the right to represent the infinite tree on the left.



The saturation process can be given in terms of what amounts to a graph rewriting system.

Definition 4.1 For M, $M' \in \mathcal{M}$ define $M \longrightarrow_{el} M'$ if there is a node u of M and there are trees S and T such that $S \leq M/u$ and $M' = M[u, T/S]$, and the trees S and T must be as depicted in one of the three cases shown below.



Define $\longrightarrow_{el}^*$ to be the transitive closure of $\longrightarrow_{el}$.

The three cases above come from the three τ laws $a \cdot x =_\tau a \cdot \tau \cdot x$ (case 1), $\tau \cdot x =_\tau \tau \cdot x + x$ (case 2), and $a \cdot (x + \tau \cdot y) =_\tau a \cdot (x + \tau \cdot y) + a \cdot y$ (case 3).

The definition amounts to stating that, $M \longrightarrow_{el} M'$ if there is an S like tree occurring in M, which replaced with the corresponding T like tree of the picture transforms M into M'. The three pictures represent three graph rewrite rules and M' is the result of applying one of those rules. The rules are not a terminating system in a strict sense, but it is the case that eventually further rewrites do not change the tree up to equivalence of $\sim$. It is a fact not proved here that for any model M there is a model M^* such that $M \longrightarrow_{el}^* M^*$ and if $M^* \longrightarrow_{el} T$ then $M^* \sim T$, moreover M^* is unique up to equivalence of $\sim$. The model M^* is a normal form for the rewrite rules of 4.1. It is not to hard to see that a normal form exists since all the rules add a new tree to the existing one and never delete anything.

The model $M(P)^*$ is the saturated model of $M(P)$. The rewrite rules of definition 4.1 are a direct translation of the saturation rewrite rules into a modal setting. If we let P^* denote the process which corresponds to the saturated transition system of P, then $M(P)^* \sim M(P^*)$. The following theorem proves that $M(P)^* \sim \mathcal{M}_{el}(P)$, so $\mathcal{M}_{el}(P)$ is a model which is the saturation of $M(P)$. Therefore $\mathcal{M}_{el}(P)$ can be identified with $M(P^*)$, and $\mathcal{M}_{el}(P)$ is the denotation of the saturated graph P^*. This explains how the mapping from CCS to $\mathcal{M}_{el}$ gives a modal semantics to the saturation process.

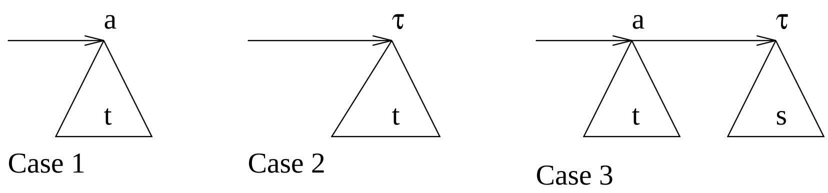
Proposition 4.2 $M(P)^* \sim \mathcal{M}_{el}(P)$

Proof Let $X = \{M \in \mathcal{M}_{el} \mid M(P) \subseteq M\}$. To prove the result it is necessary to prove the following two parts

1. $\forall M \in X,\ M(P)^* \subseteq M$
2. $M(P)^* \in X$

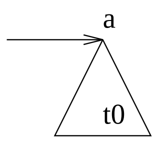
These proofs will use the tree characterising function ψ and notation defined in the proof of proposition 3.2.

Proof of part 1. Consider a tree $M \in X$ and a node v of $M(P)$. If $M(P)/v \neq 0$ then there is some $S \leq M(P)/v \subseteq M/v$ of one of the three following forms.



Consider first case 1. By applying the first rewrite rule of definition 4.1 we have $M(P) \longrightarrow_{el} M1$ where $M1 = M(P)[v, T/S]$, T being of the form of case 1 in definition 4.1. It is necessary to prove $M1 \subseteq M$.

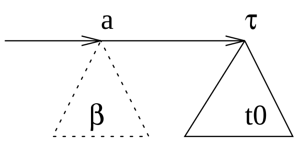
There must be some $S0 \leq M/v$ of the form



where $t \subseteq t0$. As a consequence of the first modal τ-law,

$$(v, M) \models \Diamond_a \psi(t0) \rightarrow \Diamond_a \Diamond_\tau \psi(t0)$$

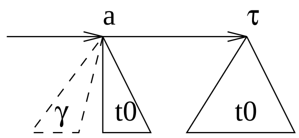
Thus there is an $S1 \leq M/v$ of the form



Let u be the node labeled with a and w be the node labeled with τ. Then by the application of the second modal τ-law,

$$(u, M) \models \psi(t0) \oplus \Diamond_\tau \psi(t0)$$

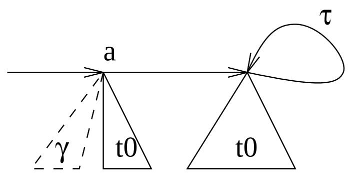
Thus $S1$ is of the form



Let $\Diamond_\tau^n f$ denote $\Diamond_\tau \Diamond_\tau \cdots \Diamond_\tau f$, the application n times of $\Diamond_\tau$ to f. From the first modal τ-law for an arbitrary n

$$(u, M) \models \Diamond_\tau \psi(t0) \leftrightarrow \Diamond_\tau^n \psi(t0)$$

Since $S1$ is a finitely branching tree this can only be true if $S1$ is of the form

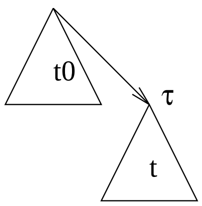


From this it follows that $M1 \subseteq M$.

The two other cases are very straightforward and show that if $M(P) \longrightarrow_{el} M1$ by either of the other two rewrite rules then $M1 \subseteq M$. This proves that if $M(P) \longrightarrow_{el}^* M1$ then $M1 \subseteq M$, hence $M(P)^* \subseteq M$. That concludes the proof of part 1.

The proof of part 2 is straightforward. For this part use the alternative set of axioms f_1 and f_2 from proposition 3.2 instead of the modal τ-laws. Take f_1 of $\mathcal{M}_{el}$, which is $\Diamond_a \Diamond_\tau x \leftrightarrow \Diamond_a x$. It is necessary to prove that $M(P)^*$ satisfies this formula. Suppose that $(v, M(P)^*) \models \Diamond_a f$ for some $f \in \mathbf{K}^\oplus$ and some node v of $M(P)^*$. Then there is an $S \leq M(P)^*/v$ of the form depicted above by case 1 from part 1 of the proof. By definition of $M(P)^*$, S must also be of the same form as depicted by $S1$ above. That is S has a single child t which is a tree $t0$ plus a τ loop at its root.

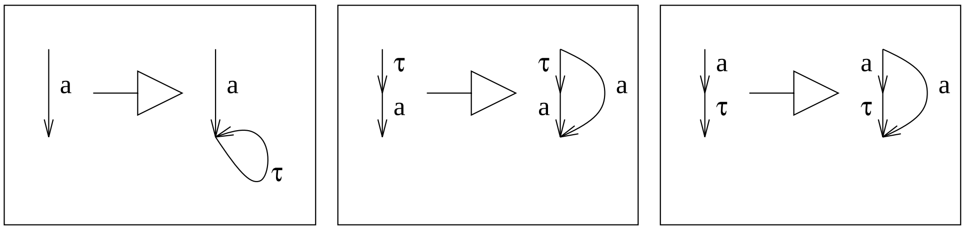
Since $(v, S) \models \Diamond_a f$ it follows that for root node u of tree t, $(u, t) \models f$. However tree t has the form of a tree $t0$ with a τ loop at its root, so that t is of the form



Hence $(u, t) \models \Diamond_\tau f$ and therefore $(v, S) \models \Diamond_a \Diamond_\tau f$. Therefore $(u, M(P)^*) \models \Diamond_a \Diamond_\tau f$ and $M(P)^* \models \Diamond_a f \rightarrow \Diamond_a \Diamond_\tau f$. The proof works the same for the implication in the other direction so that $M(P)^*$ satisfies f_1. The f_2 case is proved in an analogous way. This completes the proof.

Proof of 3.5

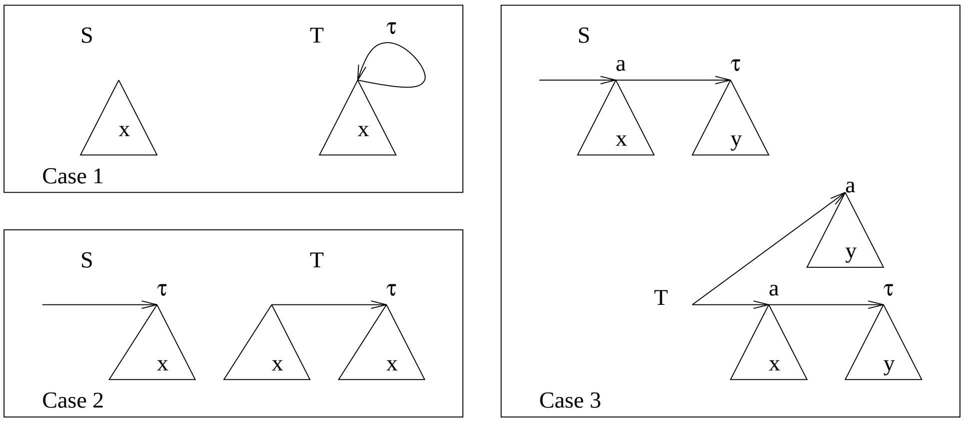
The saturation procedure of [**2**] can be described as follows. For a process P derive its transition system, i.e. the edge labeled graph which represents the derivations for P. To this graph apply the rewrite rules below exhaustively to construct the saturated graph P^* of P.



The rules are defined over edges. Thus whenever an edge or sequence of edges is found as on the left hand side of a rule, add the new edge as given by the right hand side of the rule.

Looking at the rules above it is possible to see that if they are converted into rules for trees then they give the rules of definition 4.1. Therefore $M(P^*) \sim M(P)^*$, i.e. $M(P^*) \sim \mathcal{M}_{el}(P)$. In [**2**] they prove $P =_\tau Q$ if and only if $P^* \sim Q^*$ ($\sim$ is strong bisimulation equivalence between processes). Therefore $P =_\tau Q$ iff $M(P^*) \sim M(Q^*)$ iff $M(P)^* \sim M(Q)^*$ iff $\mathcal{M}_{el}(P) \sim \mathcal{M}_{el}(Q)$. This completes the proof for part 2 of theorem 3.5.

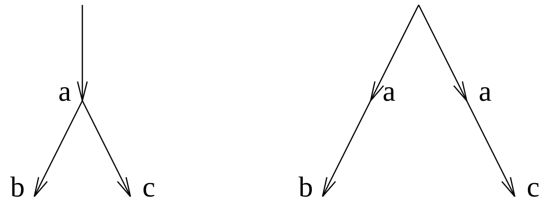
The proof for part 1 of theorem 3.5 is analogous to part 2. The following pairs when swapped with the corresponding pairs of definition 4.1 provide a definition for model $M(P)^*$ in $\mathcal{M}_{ev}$.



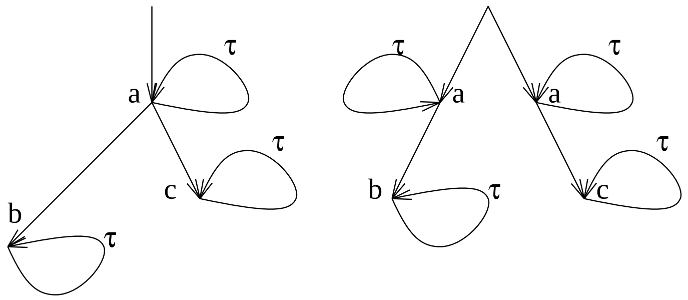
The only difference from definition 4.1 is in rule 1. For $\mathcal{M}_{ev}$ a τ loop is added to every node, not just nodes at the end of an edge. This mirrors the fact that in $\mathcal{M}_{ev}$ axiom 5 is $\mathcal{M}_{ev} \models x \rightarrow \Diamond_\tau x$. The saturation rules for $\approx$ only differ from the $=_\tau$ rules above in that the first rule states that a τ loop is added to every node. The details of the remainder of the proof are left to the reader. All the proofs go through with minor modification only.

Example

Here is a very simple example to show how processes are mapped into $\mathcal{M}_{el}$. The processes $P = a \cdot (b \cdot 0 + c \cdot 0)$ and $Q = a \cdot b \cdot 0 + a \cdot c \cdot 0$ have observation tree as depicted below in order from left to right



The trees $M(P)^*$ and $M(Q)^*$ are given below, recall they are unique only up to $\sim$ equivalence.

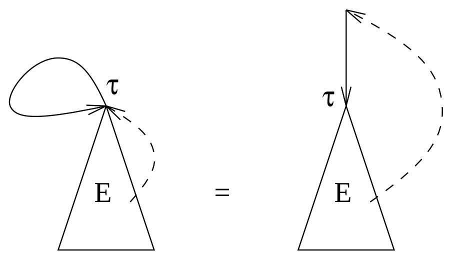


It is quite clear just by looking that these two models can not be made $\sim$ equivalent. It is also easy to prove this via the logic. In the case of process P, $M(P)^* \models_0 \boxbar_a(\boxbar_b \oplus \boxbar_c \oplus \boxbar_\tau)$ where as for Q, $M(Q)^* \not\models_0 \boxbar_a(\boxbar_b \oplus \boxbar_c \oplus \boxbar_\tau)$. Therefore $M(P)^*$ and $M(Q)^*$ are different models of the logic $\mathcal{M}_{el}$, hence P and Q are not in the same $=_\tau$ equivalence class.

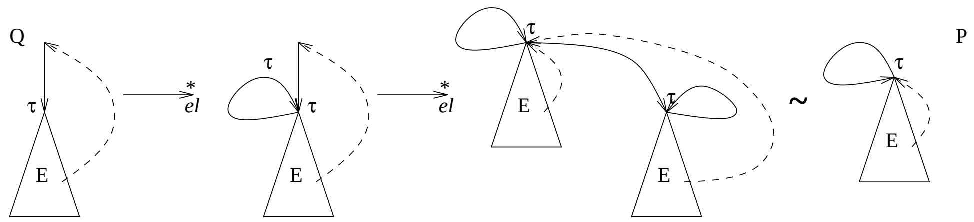
Here is a slightly more difficult example. One of the axioms for fix point processes in [**8**] is

$$\mu X(\tau \cdot X + E) =_\tau \mu X(\tau \cdot E)$$

where E is some process which involves the free variable X. Pictorially this is



Let $P = \mu X(\tau \cdot X + E)$ and $Q = \mu X(\tau \cdot E)$. The following diagram demonstrates that $Q \longrightarrow^*_{el} P$, and thus $M(Q)^* \sim M(P)^*$, proving that $P =_\tau Q$.



5 Conclusion

Modal logics $\mathcal{M}_{ev}$ and $\mathcal{M}_{el}$ have been defined to characterise $\approx$ and $=_\tau$. These logics are extensions of $\mathbf{K}^\oplus$ through the addition of simple axioms which express properties of τ. In the case of $\mathcal{M}_{el}$ the axioms are a modal representation of the τ-laws. The axioms restrict the class of models which characterise the logic. A process is mapped to the smallest model of the respective logic which contains the observation tree of the process. It has been proved that such a model is given by the saturation process of [**2**]. Two processes are equivalent if and only if they are mapped to the same model (up to equivalence of $\sim$). The saturation method explains process equality as a geometrical interpretation of the τ-laws. The logic $\mathcal{M}_{el}$ gives this geometric interpretation a modal logic semantics.

Perhaps it will be possible to phrase the π calculus [**9**] in terms of some first order extension of $\mathbf{K}^\oplus$. Such an extension may be a 'modalisation' of action structures [**10**]. Already work is being done by Hennessy and Liu [**4**], [**6**][1] to find a multi-modal logic for the π calculus. Hopefully an axiomatic approach will lead to a generalisation of saturation in a natural way.

Acknowledgements

Many thanks go to Dr Brian Monahan and Professor Howard Barringer for various spontaneous remarks which have greatly influenced my thoughts.

References

[1] P. Aczel, 'Final Universes of Processes', to appear in Proceedings of MFPS 1993.

[2] J. A. Bergstra, J. W. Klop 'A Complete Inference System for Regular Processes with Silent Moves', Proceedings of Logic Colloquium '86, Studies in Logic and the foundations of Mathematics, vol 124, North-Holland 1988.

[3] S. Graf, J.Sifakis, 'A modal characterisation of observational equivalence of finite terms of CCS', Information and Control 68, 125-145, 1986.

[4] M. Hennessy 'A Model for the π-Calculus, Technical Report, Computer Science Dept, Sussex University 1992.

[1] The title of this paper is a little misleading, this paper defines a multi-modal logic.

[5] M. Hennessey, R. Milner 'On observing nondeterminism and concurrency', Springer-Verlag LNCS 85, 295-309, 1980.

[6] M. Hennessy, X. Liu 'A Modal Logic for Message Passing Processes', Technical Report, Computer Science Dept, Sussex University January 1993.

[7] G. E. Hughes, M.J.Cresswell, 'An Introduction to Modal Logic', Methuen(UP) 1969.

[8] R. Milner 'Communication and Concurrency', Prentice Hall, 1989.

[9] R. Milner J. Parrow, D. Walker 'A Calculus of Mobile Processes' I and II, Information and Computation, V100, pp 1-78, 1992.

[10] R. Milner 'Action Structures', Technical Report, Computer Science Dept, Edinburgh University, December 1992.

[11] D. Walker 'Logics for Processes', Proceedings of the VI Banff Workshop: Process Algebras, Banff, Canada, August 1992.