A Complete Axiomatisation for Trace Congruence of Finite State Behaviors

Alexander Rabinovich
Department of Computer Science
The University of Texas at El Paso
El Paso TX 79968
e.mail: alex@cs.ep.utexas.edu

1 Introduction

Salomaa in [4] presented a complete axiomatisation for language equivalence of regular expressions. In [2] Milner proposed a complete proof system for bisimulation equivalence over finite state behaviors. Bisimulation equivalence is a very discriminating equivalence. Later, a weaker equivalence, called observational congruence, was considered by Milner and its complete axiomatisation was provided in [3]. Unlike language equivalence, both bisimulation equivalence and observational congruence distinguish between finite state behaviors on the basis of their branching structure.

In this paper trace congruence is considered. Trace congruence ignores branching structure and is close to language equivalence of classical automata theory. We provide a complete proof system for trace congruence over finite state behaviors presented as μ-expressions.

The paper is organized as follows: In Section 2 finite state behaviors are described as μ-expressions. Section 3 introduces trace equivalence. Unfortunately, trace equivalence is not substitutive. Section 3 provides a characterization of the fully abstract refinement of trace equivalence which we call trace congruence. A proof system for trace congruence is presented in section 4. The proof of its completeness is given in section 6. Section 5 contains some definitions which are helpful for the completeness proof.

In section 7 we comment about the relationship between our axiomatisation and Salomaa's classical axiomatisation of language equivalence for regular expressions. One of the open questions mentioned in [2] is to find an axiomatisation of bisimulation equivalence for finite state behaviors presented as regular expressions. We provide such an axiomatisation for a variant of regular expressions. Two appendixes contain the proofs of technical lemmas.

2 μ-Expressions

We are dealing in this paper with finite state behaviors presented as μ-expressions. Let us first recall some definitions and facts about μ-expressions and their behaviors. The presentation is based on [2, 3].

We presuppose two fixed sets

$$\begin{array}{rl} Act = \{a,\ a_1, \ldots b,\ b_1, \ldots\} & \text{the actions} \\ Var = \{X,\ X_1, \ldots Y,\ Y_1, \ldots\} & \text{the variables.} \end{array}$$

μ-expressions are defined by the following grammar:

$$E ::= 0|X|aE|E+E|\mu X.E, \text{ where } X \in Var,\ a \in Act.$$

μ stands for recursion (binding the variable which follows it). The notions of *free* and *bound* occurrences of a variable in an expression are defined as usual. An occurrence of X is *guarded* in E if it occurs within subexpression aF of E. A variable is guarded in E if all its occurrences in E are guarded.

We write $E\{E_1/X_1, \cdots E_n/X_n\}$ for the expression obtained by simultaneous substitution of E_i for each free occurrence of X_i in E, renaming bound variables as necessary. The definition is standard and is omitted.

An expression can evolve to another expression by performing an action. We write $E \xrightarrow{a} E'$ if E can evolve to E' by performing action a. The definition of the transition relation $\xrightarrow{a}$ is provided by the following inference rules:

Definition 1 *$E \xrightarrow{a} E'$ if it can be shown by the following inference rules:*

1. $aE \xrightarrow{a} E$
2. $\dfrac{E_1 \xrightarrow{a} E_2}{E_1+E \xrightarrow{a} E_2} \qquad \dfrac{E_1 \xrightarrow{a} E_2}{E+E_1 \xrightarrow{a} E_2}$

3. $\frac{E\{\mu X.E/X\}\xrightarrow{a}E_1}{\mu X.E\xrightarrow{a}E_1}$

Charts generalize automata and are defined as follows:

Definition 2 *A* chart C *is a quadruple* $< Q,\ s,\ D,\ E >$ *where:*

$$Q \text{ is a nonempty set (the nodes)}$$
$$s \in Q \text{ (the initial state)}$$
$$D \subset Q \times Act \times Q \text{ (the derivations)}$$
$$Ex \subset Q \times Var \text{ (the extensions)}$$

C *is* finite *if* $Q,\ D,\ Ex$ *are finite.*

With μ-expression E the chart (notations $Chart(E)$) is associated as follows:
Nodes: all μ-expressions.
The initial node: E.
$(E_1,\ a,\ E_2) \in D$ if $E_1 \xrightarrow{a} E_2$.
$(E, X) \in Ex$ if there is a free occurrence of X in E, which is not guarded.

Let a be an action and $s,\ s'$ be states of chart C; we shall write $s \xrightarrow{a} s'$ if $(s,\ a,\ s') \in D$.

Charts are too concrete objects. Usually a behavior equivalence $\sim$ is introduced on charts and the objects are $\sim$-equivalences classes of charts. One of the most important equivalences studied in concurrency is bisimulation equivalence $\sim_{bis}$. Milner showed

Fact 2.1 *(μ-expressions represent finite behaviors)*

1. *For every finite chart C there exists an expression E such that $Chart(E) \sim_{bis} C$.*

2. *For every expression E there exists a finite chart C such that $Chart(E) \sim_{bis} C$.*

Bisimulation equivalence is a very discriminating equivalence and most equivalences studied in the literature are coarser than it. Clearly, fact 2.1 holds when $\sim_{bis}$ is replaced by any equivalence which is coarser than bisimulation. All these justify the use of an adjective 'finite state behavior' with μ-expressions.

3 Trace Equivalence and Trace Congruence

In this section we define trace equivalence on charts. Then, we find a fully abstract refinement of trace equivalence wrt the operations: sum, prefixing, substitution and recursion.

Definition 3 *(Traces and generalized traces.) Let C be a chart and s_0 be its initial state.*

- *A trace of C is a sequence $a_1 \ldots a_n$ of actions such that there exist nodes $s_1 \ldots s_n$ in C and $s_{i-1} \xrightarrow{a_i} s_i$ for $i = 1 \ldots n$.*
- *A generalized trace of chart C is a pair consisting of a sequence $a_1 \ldots a_n$ of action and a variable X such that there exist nodes $s_1 \ldots s_n$ in C, and $s_{i-1} \xrightarrow{a_i} s_i$ for $i = 1 \ldots n$ and $X \in Ex(s_n)$.*

Remark: (1) The set of traces of chart C is a subset of Act^*; the set of generalized traces of C is a subset of $Act^* \times Var$. (2) The set of traces of C is prefix closed, i.e., if $a_1 \ldots a_n$ is a trace of C then for every $m \leq n$ its prefix $a_1 \ldots a_m$ is a trace of C. In particular, every chart contains the empty trace ϵ.

We say that expressions E and E' are trace equivalent if their corresponding charts have the same set of traces. For an expression E, we denote by $trace(E)$ the set of traces of the corresponding chart.

Lemma 3.1 *(Operations on traces)*

1. *$trace(0) = \{\epsilon\}$ - the empty string.*
2. *$trace(X) = \{\epsilon\}$ - the empty string.*
3. *$trace(E + E') = trace(E) \cup trace(E')$.*
4. *$trace(aE) = \{as \ : \ s \in trace(E)\}$.*

Unfortunately, trace equivalence is not a congruence wrt substitution and the μ operator. For example, 0 and X have the same traces, but $0\{a0/X\}$ and $X\{a0/X\}$ have different traces. We are looking for the maximal equivalence which refines trace equivalence and is a congruence wrt prefixing, sum, substitution and fixed point. Such an equivalence is called the **fully abstract** refinement of traces wrt the above operations.

Theorem 3.2 *(Full abstractness.) The fully abstract refinement of trace equivalence with respect to sum, prefixing, substitution and recursion is characterized as follows: E and E' are equivalent iff they have the same set of traces and the same set of generalized traces.*

We will use the term 'trace congruence' (notation $\sim_{trace}$) for this fully abstract equivalence.

Proof: The proof that this equivalence is a congruence is quite straightforward and is omitted.

To show that it is fully abstract, assume that the set of variables which are free in E and E' is a subset of $\{X_1, \ldots X_n\}$. Let $a_1 \ldots a_n$ be different actions which do not appear in E, E'. Then $E\{a_1 0/X_1 \ldots a_n 0/X_n\}$ and $E'\{a_1 0/X_1 \ldots a_n 0/X_n\}$ are trace equivalent only if E and E' have the same set of generalized traces. Therefore, if E and E' are related by a congruence $\sim$ which refines trace equivalence, they have the same set of generalized traces. Since $\sim$ refines trace equivalence, E and E' should also have the same set of traces.

To summarize, $\sim_{trace}$ is a congruence and any congruence $\sim$ which refines trace equivalence should refine $\sim_{trace}$. Therefore, $\sim_{trace}$ is the fully abstract refinement of trace equivalence. $\square$

Remark: (1) Trace congruence is also the fully abstract refinement of trace equivalence wrt to the operation prefix, sum and recursion. (2) If every node of charts C, C' has the empty extension, then C and C' are trace congruent iff they are trace equivalent. In particular, trace congruence and trace equivalence coincide for μ-expressions without free variables.

4 Proof System

Our proof system for trace congruence in addition to the standard equivalence and congruence inference rules has the following axioms and fixed point inference rule.

Axioms

S1 $E + F = F + E$

S2 $(E + F) + G = E + (F + G)$

S3 $E + E = E$

S4 $E + 0 = E$

R1 $\mu X.\ E = \mu Y.(E\{Y/X\})$, Y not free in E

R2 $\mu X.\ E = E\{\mu X.\ E/X\})$

R3 $\mu X.\ E = \mu X.\ (E + X)$

P1 $aX + aY = a(X + Y)$

Fixed point Inference Rule

R4 From $E = F\{E/X\}$, X guarded in F, infer $E = \mu X.\ F$.

The set of axioms *S1-S4* is sound and complete for bisimulation equivalence over the recursive free subset of μ-expressions [1]. It is well known that by augmenting this set by *P1* a sound and complete system is obtained for trace equivalence over the variable free subset of μ-expressions. Moreover, it is not difficult to check that *S1-S4, P1* are also sound and complete for trace congruence over the recursive free subset of μ-expressions.

The main result of [2] is that the set of axioms *S1-S4, R1-R3* and fixed point inference rule *R4* is a sound and complete system for bisimulation equivalence. Our system is obtained by augmenting this set by *P1*; we will show that it is sound and complete for trace congruence.

Notations: We write $\vdash E = E'$ if $E = E'$ is provable in our system. We write $\vdash_M E = E'$ if $E = E'$ is provable without using axiom *P1*, i.e. it is provable in Milner's system.

5 Systems of Equations

This section contains some definitions which are needed for the completeness proof given in the next section.

Definition 4 *(an $(\tilde{X};\ \tilde{Y})$ system of equations) Let $\tilde{X} = \{X_1,\ \ldots X_n\}$, $\tilde{Y} = \{Y_1,\ \ldots Y_m\}$ be different variables and $\tilde{F} = \{F_1 \ldots F_n\}$ expressions with free variables in $\tilde{X} \cup \tilde{Y}$. A sequence of formal equations $Sys :=< X_1 = F_1, \ldots X_n = F_n >$ is called an $(\tilde{X};\ \tilde{Y})$ system of equations; $\tilde{X}$ are called the bound variables of the system and X_1 is the principal variable of the system.*

Definition 5 *(Guarded, standard and deterministic systems)*

- *An $(\tilde{X};\ \tilde{Y})$ system is* guarded *if all $\tilde{X}$ variables are guarded in the expressions F_i of the system.*
- *A system is* standard *if all F_i are of the form $\sum_{j \in I_i} a_{i,j} X_{f(i,j)} + \sum_{j \in K_i} Y_{g(i,j)}$.*
- *A standard system is called* deterministic *if $a_{i,j} = a_{i,j'}$ implies that $j = j'$.*

Note that standard and deterministic systems of equations are guarded.

Definition 6 *(Solutions of a system) Consider an $(X_1 \ldots X_n; Y_1, \ldots Y_m)$ system Sys with equations $X_i = F_i$, for $i = 1, \ldots n$.*

- *A sequence* $E_1,\ E_2, \ldots E_n$ *of expressions is a solution of* Sys *iff for every* i $E_i \sim_{trace} F_i\{E_1/X_1, \ldots E_n/X_n\}$*; expression* E_1 *is called a principal solution of the system.*
- $E_1,\ E_2, \ldots E_n$ *is an M-provable solution of* Sys *iff* $\vdash_M E_i = F_i\{E_1/X_1, \ldots E_n/X_n\}$ *for every* i*; expression* E_1 *is called a principal M-provable solution of the system.*
- $E_1,\ E_2, \ldots E_n$ *is a provable solution of* Sys *iff* $\vdash E_i = F_i\{E_1/X_1, \ldots E_n/X_n\}$ *for every* i*; expression* E_1 *is called a principal provable solution of the system.*

6 Soundness and Completeness

Theorem 6.1 *(Soundness) The axioms and the inference rules are sound for trace congruence.*

As usual, the proof of soundness theorem is simple and we concentrate here on the completeness proof. Our completeness proof is based on Milner's completeness proof for bisimulation equivalence [2]. Many of our arguments are modifications of his ideas.

The following theorem was proved by Milner (theorem 5.7 in [2]).

Theorem 6.2 *(Unique M-provable solution of equations.) Every guarded* $(\tilde{X};\ \tilde{Y})$ *system of equations* Sys *has a unique M-provable solution, i.e.,* Sys *has an* M*-provable solution and if both* $E_1 \cdots E_n$ *and* $E'_1 \cdots E'_n$ *are* M *provable solutions of* Sys *then* $\vdash_M E_i = E'_i$ *for* $i = 1, \cdots, n$*.*

In [2] it was shown (theorem 5.8)

Theorem 6.3 *(Equational characterization of* μ*-expressions.) Every* μ*-expression is a principal M-provable solution of a standard system of equations.*

M-provable equations are provable, therefore,

Corollary 6.4 *Every* μ*-expression is a principal provable solution of a standard system of equations.*

We strengthen the corollary and show

Theorem 6.5 *Every* μ*-expression is a principal provable solution of a deterministic system of equations.*

Proof: The proof is given in appendix A. □

Our proof of the completeness theorem uses the following

Claim 6.6 *Assume (1) $E \sim_{trace} E'$ (2) E is a provable principal solution of deterministic system of equations Sys (3) E' is a provable principal solution of deterministic system of equations Sys'. Then there exists a guarded system Sys'' such that both E and E' are its principal provable solutions.*

Proof: The proof is given in appendix B. □

Theorem 6.7 *(Completeness) If $E \sim_{trace} E'$ then $\vdash E = E'$.*

Proof: Actually, Milner's proof of theorem 6.2 also gives the *unique provable solution theorem*: every guarded system of equations has a unique provable solution.

Now, by theorem 6.5 there exist deterministic systems of equations Sys and Sys' such that E and E' are principal provable solutions of Sys and Sys' respectively.

Therefore, by claim 6.6, there exists a guarded system of equations Sys'' such that both E and E' are its principal provable solutions. Therefore, by the unique provable solution theorem, $\vdash E = E'$. □

7 Relationship to Salomaa's axiomatisation

In [4] Salomaa presented a complete axiom system for language equivalence over regular expressions. Trace congruence is conceptually close to language equivalence. Our proof has the same structure as Salomaa's (unique provable solution theorem, equational characterization theorem), but in many technical arguments it is closer to Milner's proof for bisimulation equivalence and surprisingly, we were unable to adopt Salomaa's proof for a complete axiomatisation of trace congruence.

The main obstacle for extending Salomaa's proof lies in the fact that the empty language is present explicitly in his axiomatisation and plays a very important role there. However, for trace congruence there is no expression which 'corresponds' to the empty language. In particular, unlike the language law $b\emptyset = \emptyset = a\emptyset$, there exists no expression E for which aE is trace congruent to bE.

In the rest of this section we consider an interesting subset of regular expressions. Adopting Salomaa's results, we provide an axiomatisation of bisimulation equivalence and an axiomatisation of trace congruence over this subset. The proofs are omitted and will be given in the full paper.

7.1 #-Expressions

In remark 5 of [4] it is explained how to axiomatise 'regular expressions' without the empty language.

More exactly, Salomaa considers expressions constructed from an alphabet by the following operations: concatenation, sum and positive iteration # ($A^{\#} = A + AA + AAA + \ldots$). Let us call such expressions - #-expressions. It is claimed there that the following proof system is complete for #-expressions:

Axioms:

A1:	Associativity of sum.
A2:	Associativity of concatenation.
A3:	Commutativity of sum.
A4:	$A(B + C) = AB + AC$.
A5:	$(A + B)C = AB + AC$.
A6:	$A + A = A$.
A'10:	$A^{\#} = A + AA^{\#}$

Inference rules:

R1:	Usual Congruence rules.
R2:	$A = BA + C$ implies $A = B^{\#}C + C$.

7.2 Embedding of #-expressions into μ-expressions

Similar to Milner's [2] embedding of the standard regular expressions into μ-expressions one can embed #-expressions into μ-expressions.

Let X be a distinguished variable. Define:

1. $Em(a) = aX$ for every action a.
2. $Em(A + B) = Em(A) + Em(B)$.
3. $Em(AB) = Em(A)\{Em(B)/X\}$.
4. $Em(A^{\#}) = \mu Y.A + A\{Y/X\}$

Remark 1. In the chart of $Em(A)$ the nodes may have either an extension X or the empty extension. From every node, a node with extension X is reachable. The initial node has the empty extension. Such a chart can be considered as an automaton whose accepting states are the nodes with extension X. The language accepted by this automaton coincides with the language defined by expression A.

Remark 2. (Axiomatisation of trace congruence over #-expressions.) This embedding is adequate for trace congruence, i.e., $A = B$ is provable in Salomaa's axiomatisation iff $Em(A) = Em(B)$ is provable in our axiomatisation of trace congruence.

7.3 Axiomatisation of #-expression wrt bisimulation

Bisimulation equivalence, due to Milner and Park, is one of the most fundamental equivalences that has emerged in concurrency. Its definition is omitted here, but let us note that axiom A'10 holds even for bisimulation equivalence, i.e. $Em(A^{\#})$ is bisimilar to $Em(A) + Em(A)Em(A^{\#})$. It is also easy to see that the axioms A1, A2, A3, A5, A6 and the inference rules R1, R2 are valid for bisimulation, but axiom A4 fails.

We do not know whether the set A1, A2, A3, A5, A6, A'10, R1, R2 is complete for bisimulation equivalence over #-expressions. One can show that there exists no #-expression which solves equation $Y =_{bis} aY + b$. We do not know how to characterize the systems of equations with the solutions bisimilar to #-expressions.

However, we can prove that a complete proof system for bisimulation equivalence is obtained when R2 is replaced by the following inference rule:

R3: Every system of equations of the form $Y_i = \sum a_{i,j} Y_{f(i,j)} + \sum b_{i,j}$ has at most one solution.

8 Further Results

(a) Axiomatisation of trace *approximation* - straightforward.

(b) τ -action. The addition of axiom $E = \tau E$ provides a complete axiomatisation of trace congruence with unobservable action τ.

(c) *Divergence.* Trace congruence identifies expressions 0 and $\mu X.\ X$. We can introduce the notion of divergence and provide a complete axiomatisation for the refinement of trace congruence which properly takes into account divergence.

Acknowledgments

Our completeness proof is based on Milner's proof [2]. I am grateful to Robin Milner for his remarks on the draft of the paper. The results of section 7 were inspired by these comments.

References

[1] M. Hennesy and R. Milner. Algebraic laws for nondeterminism and concurrency. In *J. Assoc. Comp. Mach.* , volume 32, pp. 137-161, 1985.

[2] R. Milner. A complete proof system for a class of regular behaviors. In *JCSS* , volume 28, pp. 439-466, 1984.

[3] R. Milner. A complete axiomatisation for observational Congruence of finite state behaviors. In *Information and Computation*, volume 81, pp. 227-247, 1989.

[4] A. Salomaa. Two complete axiom systems for the algebra of regular events. In *J. Assoc. Comp. Mach.*, volume 13(1) pp. 158-169, 1965.

Appendix

The appendix contains two sections. In section A theorem 6.5 is proved; in section B claim 6.6 is proved.

A Determinization

In this appendix we show that every expression is a principal provable solution of a deterministic system of equations. This fact follows immediately from the corollary 6.4 and the following

Claim A.1 *If E_1 is a principal provable solution of a standard system of equations, then E_1 is a principal provable solution of a deterministic system of equations.*

First, we introduce some notations which are helpful in the proof of the claim.

Let $X_i = \sum_{j \in J_i} a_{i,j} X_{f(i,j)} + \sum_{k \in K_i} Y_{g(i,k)}$ be an equation.

Define

$S(i) = \{a_{i,j} : \ j \in J_i\}$ - successor actions of X_i.
$D(i,a) = \{r : \ aX_r \ \textit{is a summand of the equation}\}$ - a-derivatives of X_i.
$Ex(i) = \{r : \ Y_r \ \textit{is a summand of the equation}\}$ - extensions of X_i.

Extend point-wise these functions to the subsets of $\{1, \ldots p\}$, i.e.

$S(Q) = \cup_{i \in Q} S(i)$,
$D(Q,a) = \cup_{i \in Q} D(i,a)$,
$Ex(Q) = \cup_{i \in Q} Ex(i)$.

Now, if $X_i = \sum_{j \in J_i} a_{i,j} X_{f(i,j)} + \sum_{k \in K_i} Y_{g(i,k)}$ is an equation in our system, it can be easily shown, by the axioms for sum, that

$$\vdash_M \sum_{j \in J_i} a_{i,j} X_{f(i,j)} + \sum_{k \in K_i} Y_{g(i,k)} = \sum_{a \in S(i)} (\sum_{r \in D(i,a)} aX_r) + \sum_{r \in Ex(i)} Y_r.$$

Therefore, every standard system of equations is $\vdash_M$ equivalent to a system with equations of the form $X_i = \sum_{a \in S(i)} (\sum_{r \in D(i,a)} aX_r) + \sum_{r \in Ex(i)} Y_r$. From now on, we will only consider such systems.

Now we are ready to start the proof of claim A.1.

Since E_1 is a principal provable solution of a standard system of equations there exist $E_2, \ldots E_p$ and a system of equations:

$$Sys = \begin{cases} X_1 & = & \sum_{a \in S(1)} (\sum_{r \in D(1,a)} aX_r) + \sum_{r \in Ex(1)} Y_r \\ \vdots & \vdots & \vdots \\ X_i & = & \sum_{a \in S(i)} (\sum_{r \in D(i,a)} aX_r) + \sum_{r \in Ex(i)} Y_r \\ \vdots & \vdots & \vdots \\ X_p & = & \sum_{a \in S(p)} (\sum_{r \in D(p,a)} aX_r) + \sum_{r \in Ex(p)} Y_r \end{cases}$$

such that for every i

$$\vdash E_i = \sum_{a \in S(i)} (\sum_{r \in D(i,a)} a E_r) + \sum_{r \in Ex(i)} Y_r \tag{1}$$

We are going to define a deterministic system of equations Sys' and to show that E_1 is its principal solution. The bound variables of Sys' are indexed by the subsets of $\{1, \ldots p\}$; X_1 is its principal variable. For a set $Q \subset \{1, \ldots p\}$ the equation for X_Q is:

$$X_Q = \textstyle\sum_{a \in S(Q)} a X_{D(Q,a)} + \sum_{r \in Ex(Q)} Y_r$$

It is clear that this is a deterministic system.

Define $E_Q \equiv \sum_{i \in Q} E_i$. We claim that $\vdash E_Q = \sum_{a \in S(Q)} a E_{D(Q,a)} + \sum_{r \in Ex(Q)} Y_r$.

Note that

$$\vdash E_{Q_1} + E_{Q_2} = E_{Q_1 \cup Q_2} \tag{2}$$

Recall that by the assumption of the claim

$$\vdash E_i = \sum_{a \in S(i)} (\sum_{r \in D(i,a)} a E_r) + \sum_{r \in Ex(i)} Y_r \tag{3}$$

Therefore, adding the equations for $i \in Q$

$$\vdash \sum_{i \in Q} E_i = \sum_{i \in Q} (\sum_{a \in S(i)} (\sum_{r \in D(i,a)} a E_r) + \sum_{r \in Ex(i)} Y_r) \tag{4}$$

The left-hand side is E_Q by the definition of E_Q. The right-hand side can be rearranged, by the axioms for $+$, and gives

$$\vdash E_Q = \sum_{a \in \cup_{i \in Q} S(i)} (\sum_{r \in \cup_{i \in Q} D(i,a)} a E_r) + \sum_{r \in \cup_{i \in Q} Ex(i)} Y_r \tag{5}$$

By the idempotence of $+$, and the fact that $S,\ D,\ Ex$ are extended point-wise to sets we obtain:

$$\vdash E_Q = \sum_{a \in S(Q)} (\sum_{r \in D(Q,a)} a E_r) + \sum_{r \in Ex(Q)} Y_r \tag{6}$$

Now applying the prefix axiom, $aX + aY = a(X + Y)$, we derive

$$\vdash E_Q = \sum_{a \in S(Q)} a \sum_{r \in D(Q,a)} E_r + \sum_{r \in Ex(Q)} Y_r \tag{7}$$

By the definition, $E_{D(Q,a)} \equiv \sum_{i \in D(Q,a)} E_i$, therefore

$$\vdash E_Q = \sum_{a \in S(Q)} a E_{D(Q,a)} + \sum_{r \in Ex(Q)} Y_r \tag{8}$$

Hence, the equations obtained by substitution of E_Q for X_Q in the equations of Sys' are provable. In particular, E_1 is a principal provable solution of Sys'.

B Proof of Claim 6.6

In this section we prove claim 6.6. We recall claim 6.6.

Claim B.1 *Assume (1) $E_1 \sim_{trace} E'_1$ (2) E_1 is a provable principal solution of deterministic system of equations Sys (3) E'_1 is a provable principal solution of deterministic system of equations Sys'. Then there exists a guarded system Sys'' such that both E_1 and E'_1 are its principal provable solutions.*

The proof is based on

Lemma B.2 *Assume that (1) $E \sim_{trace} G$ (2) $E \sim_{trace} \sum_{i \in I} a_i E_i + \sum_{k \in K} Y_k$ and (3) $G \sim_{trace} \sum_{j \in J} b_j G_j + \sum_{m \in M} Z_m$, where a_i are different actions and b_j are different actions. Then (a) $\{Y_k : k \in K\} = \{Z_m : m \in M\}$; (b) $|I| = |J|$ and there exists a one-one function $h : I \to J$ such that $a_i = b_{h(i)}$ and $E_i \sim_{trace} G_{h(i)}$. (c) $G \sim_{trace} \sum_{i \in I} a_i G_{h(i)} + \sum_{k \in K} Y_k$ and $\vdash \sum_{j \in J} b_j G_j + \sum_{m \in M} Z_m = \sum_{i \in I} a_i G_{h(i)} + \sum_{k \in K} Y_k$.*

Proof: The initial node s_0 of the chart for $\sum_{i \in I} a_i E_i + \sum_{k \in K} Y_k$ has the extension $\{Y_k : k \in K\}$ and its set of successor actions is $\{a_i : i \in I\}$. The initial node s'_0 of the chart for $\sum_{j \in J} b_J G_j + \sum_{m \in M} Z_k$ has the extension $\{Z_m : m \in M\}$ and its set of successor actions is $\{b_j : j \in J\}$. In equal charts the initial nodes have the same extension and the same set of successor actions, therefore $\{Y_k : k \in K\} = \{Z_m : m \in M\}$ and $\{a_i : i \in I\} = \{b_j : j \in J\}$. Moreover, since all a_i are different and all b_j are different the relation $R(i, j) =_{def} a_i \equiv b_j$ is a graph of a one-one function h between sets I and J. In particular, using associativity and commutativity of $+$, the equation for G can be rewritten into $G \sim_{trace} \sum_{i \in I} a_i G_{h(i)} + \sum_{m \in M} Z_m$ and $\vdash \sum_{j \in J} b_j G_j + \sum_{m \in M} Z_m = \sum_{i \in I} a_i G_{h(i)} + \sum_{k \in K} Y_k$.

In the case when all actions a_i are different, it can be easily shown that $(a_i s, X)$ is a generalized trace of $\sum_{i \in I} a_i E_i + \sum_{k \in K} Y_k$ iff (s, X) is a generalized trace of E_i. Similarly, $(a_i s, X)$ is a generalized trace of $\sum_{i\in I} a_i G_{h(i)} + \sum_{k\in K} Y_k$ iff (s, X) is a generalized trace of $G_{h(i)}$. Since $\sum_{i\in I} a_i E_i + \sum_{k\in K} Y_k$ and $\sum_{i\in I} a_i G_{h_i} + \sum_{k\in K} Y_k$ have the same set of generalized traces, it follows that E_i and $G_{h(i)}$ have the same set of generalized traces. Similar arguments show that E_i and $G_{h(i)}$ have the same set of traces. Hence $E_i \sim_{trace} G_{h(i)}$. □

Now we proceed with the proof of claim B.1. Let $E_1, \ldots E_p$ be a provable solution of a deterministic system of equations:

$$\vdash E_i = \sum_{j\in J_i} a_{i,j} E_{f(i,j)} + \sum_{j\in K_i} Y_{g(i,j)} \quad i = 1, \ldots p. \tag{9}$$

Let $E'_1, \ldots E'_m$ be a provable solution of deterministic system of equations:

$$\vdash E'_{i'} = \sum_{j\in J'_{i'}} b_{i',j} E'_{f'(i',j)} + \sum_{j\in K'_{i'}} Y'_{g'(i',j)} \quad i' = 1, \ldots m. \tag{10}$$

All provable equations are valid, therefore in the above equations $=$ can be replaced by $\sim_{trace}$. Define set $R = \{(i, i') : \ E_i \sim_{trace} E'_{i'}\}$. R is not empty, because $(1,1) \in R$.

For $(i, i') \in R$, all the assumptions of lemma B.2 hold, therefore there exists a one-one function $h_{i,i'} : \ J_i \to J'_{i'}$ such that $a_{i,j} = b_{i',h_{i,i'}(j)}$ and $E_{f(i,j)} \sim_{trace} E'_{f'(i',h_{i,i'}(j))}$.

Consider the following system Sys'' of equations with bound variables indexed by the elements of R. The equation for $X_{i,i'}$ is:

$$X_{i,i'} = \sum_{j\in J_i} a_{i,j} X_{f(i,j),f'(i',h_{i,i'}(j))} + \sum_{j\in K_i} Y_{g(i,j)} \tag{11}$$

Note that all bound variables in this system are indexed by elements of R. Indeed $E_{f(i,j)} \sim_{trace} E'_{f'(i',h_{i,i'}(j))}$ by the conclusion (c) of lemma B.2. Therefore, the pair $(f(i,j),\ f'(i', h_{i,i'}(j)))$ is in R.

Define a sequence of expressions indexed by elements of R: $E_{i,i'} \equiv E_i$. We claim that when $E_{i,i'}$ are substituted for $X_{i,i'}$ in Sys'', then the resulting equations are provable. Indeed, after substitution, equations (i, i') become

$$E_{i,i'} = \sum_{j\in J_i} a_{i,j} E_{f(i,j),f'(i',h_{i,i'}(j))} + \sum_{j\in K_i} Y_{g(i,j)} \tag{12}$$

and by the definition of $E_{i,i'}$ it is equivalent to

$$E_i = \sum_{j\in J_i} a_{i,j} E_{f(i,j)} + \sum_{j\in K_i} Y_{g(i,j)} \tag{13}$$

which by our assumption is provable. Therefore (12) is also provable.

Now define another sequence of expressions indexed by elements of R: $E'_{i,i'} \equiv E'_{i'}$. Again we claim that when $E'_{i,i'}$ are substituted for $X_{i,i'}$ in Sys'' then the resulting equations are provable. Indeed, after substitution, the equations for $X_{i,i'}$ become

$$E'_{i,i'} = \sum_{j \in J_i} a_{i,j} E'_{f(i,j),f'(i',h_{i,i'}(j))} + \sum_{j \in K_{i'}} Y_{g(i,j)} \tag{14}$$

and by the definition of $E'_{i,i'}$ it is provable iff

$$\vdash E'_{i'} = \sum_{j \in J_i} a_{i,j} E'_{f'(i',h_{i,i'}(j))} + \sum_{j \in K_i} Y_{g(i,j)} \tag{15}$$

By (c) of the lemma $\vdash \sum_{j \in J'_{i'}} b_{i',j} E'_{f'(i',j)} + \sum_{j \in K_{i'}} Y'_{g'(i,j)} = \sum_{j \in J_i} a_{i,j} E'_{f'(i',h_{i,i'}(j))} + \sum_{j \in K_i} Y_{g(i,j)}$ therefore (14) is provable iff the following is provable:

$$\vdash E'_{i'} = \sum_{j \in J'_{i'}} b_{i',j} E'_{f'(i',j)} + \sum_{j \in K'_{i'}} Y'_{g'(i',j)} \quad i' = 1, \ldots m. \tag{16}$$

However, this is provable by our assumption (see equation (10) above). Hence, both $E_{1,1}$ and $E'_{1,1}$ are principal provable solutions of Sys''. But, by the definition $E_{1,1} \equiv E_1$ and $E'_{1,1} \equiv E'_1$. Therefore E_1 and E'_1 are principal provable solutions of Sys''.

Finally, note that Sys'' is a guarded system and its size is polynomial in the sizes of Sys and Sys'.