Canonical Solutions to Recursive Equations and Completeness of Equational Axiomatisations

Xinxin Liu
State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, China
University of Chinese Academy of Sciences, China
xinxin@ios.ac.cn

TingTing Yu
Beijing Sunwise Information Technology Ltd, China
Beijing Institute of Control Engneering, China
yutingting@sunwiseinfo.com

Abstract

In this paper we prove completeness of four axiomatisations for finite-state behaviours with respect to behavioural equivalences at various τ-abstract levels: branching congruence, delay congruence, η-congruence, and weak congruence. Instead of merging guarded recursive equations, which was the approach originally used by Robin Milner and has since become the standard strategy for proving completeness results of this kind, in this work we take a new approach by solving guarded recursive equations with canonical solutions which are those with the fewest reachable states. The new strategy allows uniform treatment of the axiomatisations with respect to different behavioural equivalences.

2012 ACM Subject Classification Theory of computation → Process calculi

Keywords and phrases Bisimulation, Congruence, Axiomatisation, Soundness and Completeness

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2020.35

Funding *Xinxin Liu*: The work has been partially supported by the CAS-INRIA major project No. GJHZ1844, and by NSFC under grants No. 61836006 and No. 61672540.

Acknowledgements We thank the reviewers' suggestions which greatly improved the presentation.

1 Introduction

The notion of bisimulation which originated from the early ideas of Park and van Benthem and coined by Milner is the foundation of many popular equivalence and congruence relations in concurrency theory. Weak bisimilarity introduced by Milner [6] (originally called observational equivalence) and branching bisimilarity introduced by van Glabbeek and Weijland [9] are two widely studied equivalence relations for processes, with the former identifying more processes than the latter because of the difference in treating internal actions (τ-moves) to achieve observational abstractness. Delay bisimilarity and η-bisimilarity [9] are two other interesting equivalence relations with abstractness in between weak and branching bisimilarities. Although these relations are not congruences on process constructs, all can be made into congruences, called weak congruence and branching congruence etc., which are very close to the respective bisimilarities. For finite processes where every process will eventually become inactive, complete inference systems for all four congruences mentioned above can be found in the literature. For example, a complete axiomatisation of finite processes with respect to weak congruence was devised by Hennessy and Milner [3], and complete axiomatisations of finite processes with respect to branching, delay, and η-congruences were devised by van Glabbeek and Weijland [9]. For finite-state processes where recursion is allowed in the process constructs to introduce infinite behaviours, complete inference systems for weak congruence and branching congruence were devised by Milner [7] and van Glabbeek [8] respectively, while such inference systems for the other two congruences are not found in the literature. In this paper we present a hierarchy of complete inference systems for finite-state processes with respect to the four congruences. For proving completeness, we use a new approach by constructing canonical solutions to guarded recursive equations which is different from the common strategy of merging guarded recursive equations, a strategy originally proposed by Milner [5] and later followed in many other completeness proofs [7][10][8][4]. Our new strategy allows more uniform treatment of the axiomatisations with respect to different behavioural equivalences. In the following we roughly explain the ideas of the old and new approaches.

Both the old and new approaches rely on the following two fundamental facts:

Fact 1. Each expression provably solves a set of guarded recursive equations.
Fact 2. A set of guarded recursive equations has unique solution in the following sense: if two expressions both provably solve the same set of guarded recursive equations, then the two expressions are provably equal.

In [5], after setting up the foundation by proving the two facts above, Milner then devised the following strategy to establish the provable equality of two semantically equivalent expressions E_1 and E_2:

Step 1. By Fact 1. above, let S_1, S_2 be two sets of guarded recursive equations such that E_1 and E_2 provably solve S_1 and S_2 respectively.
Step 2. Then, with the help of the semantic equality of E_1 and E_2, by merging S_1 and S_2 in a systematic way to form a single set of guarded recursive equations S which is provably solved by both E_1 and E_2.
Step 3. Then by Fact 2. above, it follows that E_1 and E_2 are provably equal.

For Step 2. Milner provided a procedure which guarantees that the wanted guarded recursive equation set S can be constructed. To see a tiny but nevertheless illustrative example, consider the expressions $\mu X.(a.a.X)$ and $\mu Y.(a.a.a.Y)$, they both perform the visible action a forever. If we write $\vdash E = F$ to mean that the expressions E and F are provably equal, then following Milner's strategy, first note that by unfolding recursion we obtain

$$\vdash \mu X.(a.a.X) = a.a.\mu X.(a.a.X), \quad \vdash \mu Y.(a.a.a.Y) = a.a.a.\mu Y.(a.a.a.Y).$$

That is, $\mu X.(a.a.X)$ and $\mu Y.(a.a.a.Y)$ provably solve $\{X = a.a.X\}$ and $\{Y = a.a.a.Y\}$ respectively. Now by equational reasoning, if $\vdash E = a.a.E$, then $\vdash a.a.E = a.a.a.a.E$, and $\vdash a.a.a.a.E = a.a.a.a.a.a.E$, thus $\vdash E = a.a.a.a.a.a.E$. That is to say, a solution to $X = a.a.X$ also provably solves $Z = a.a.a.a.a.a.Z$. And for the same reason a solution to $Y = a.a.a.Y$ also provably solves $Z = a.a.a.a.a.a.Z$. Thus both $\mu X.(a.a.X)$ and $\mu Y.(a.a.a.Y)$ provably solve $\{Z = a.a.a.a.a.a.Z\}$, hence by Fact 2. $\vdash \mu X.(a.a.X) = \mu Y.(a.a.a.Y)$. Note that here we have simplified Milner's procedure by allowing sets of guarded equations which may not be in the *standard form* required in his procedure. That is sufficient to make our points. (The sets of guarded equations in standard form which provably solved by $\mu X.(a.a.X)$ and $\mu Y.(a.a.a.Y)$ are $\{X_0 = a.X_1, X_1 = a.X_0\}$ and $\{Y_0 = a.Y_1, Y_1 = a.Y_2, Y_2 = a.Y_0\}$ respectively, which can be merged to obtain the following set of standard equations, provably solved by both $\mu X.(a.a.X)$ and $\mu Y.(a.a.a.Y)$ in Z_{00}: $\{Z_{00} = a.Z_{11}, Z_{11} = a.Z_{02}, Z_{02} = a.Z_{10}, Z_{10} = a.Z_{01}, Z_{01} = a.Z_{12}, Z_{12} = a.Z_{00}\}$.)

Now by examining Milner's approach, we can find an interesting alternative which has never been explored. The idea, which is very simple, is as follows. After Step 1., instead of merging S_1 and S_2 to obtain S in Step 2. (which is the most involving and complex step in this approach), is it possible to find *an expression* E which provably solves both S_1 and S_2? If the answer is yes, then by Fact 2. E is provably equal to both E_1 and E_2, thus by transitivity E_1 is provably equal to E_2. Let us apply this alternative approach to the above example, and consider the expression $\mu Z.(a.Z)$. By unfolding we obtain $\vdash \mu Z.(a.Z) = a.\mu Z.(a.Z)$, and then by equational reasoning we obtain $\vdash a.\mu Z.(a.Z) = a.a.\mu Z.(a.Z)$ and $\vdash a.a.\mu Z.(a.Z) = a.a.a.\mu Z.(a.Z)$, thus $\vdash \mu Z.(a.Z) = a.\mu Z.(a.Z) = a.a.\mu Z.(a.Z) = a.a.a.\mu Z.(a.Z)$. This shows that, $\mu Z.(a.Z)$ provably solves both $\{X = a.a.X\}$ (which is also probably solved by $\mu X.(a.a.X)$) and $\{Y = a.a.a.Y\}$ (which is also provably solved by $\mu Y.(a.a.a.Y)$). Then we can use Fact 2. to obtain $\vdash \mu Z.(a.Z) = \mu X.(a.a.X)$ and $\vdash \mu Z.(a.Z) = \mu Y.(a.a.a.Y)$, hence $\vdash \mu X.(a.a.X) = \mu Y.(a.a.a.Y)$.

The work of this paper is to show that the alternative approach works in general case. We show that the common provable solution E to S_1 and S_2 can always be constructed out of canonical solutions to some set of guarded recursive equations, where canonical solutions are those with the smallest number of reachable states amongst the solutions.

The paper is organized as follows. In the next section we settle the preliminaries. In section 3 we present a hierarchy of axiomatisations for branching congruence, delay congruence, η-congruence, and weak congruence, and state some basic properties. In section 4 we carry out in detail the construction of canonical solutions to recursive equations to prove the completeness result for branching congruence. In section 5 we prove some saturation results, and with which to obtain the completeness with respect to other three congruences. Then, after discussing related work in section 6, we conclude in section 7.

2 Expressions, Equivalences, and Congruences

Let $\mathcal{V}$ be an infinite set of *variables*, $\mathcal{A}$ be an infinite set of *visible actions*, τ be the *invisible action* or *silent move* ($\tau \notin \mathcal{A}$). We write $\mathcal{A}_\tau$ for $\mathcal{A} \cup \{\tau\}$. The set $\mathcal{E}$ of *regular process expressions* is given by the following BNF grammar:

$$E \quad ::= \quad \mathbf{0} \;\Big|\; X \;\Big|\; a.E \;\Big|\; E + E \;\Big|\; \mu X.E$$

where $a \in \mathcal{A}_\tau$, $X \in \mathcal{V}$. Here $\mathbf{0}$ is the null expression which is not capable of any action; $a.E$ is a prefix expression which first performs the action a and then proceeds as E; $E + F$ is a summation which will behave as either E or F; $\mu X.E$ stands for recursion, with μ binding the variable X in E. We assume the usual notion of free and bound occurrences of variables with respect to the variable binder μ, and write $E\{F/X\}$ for the (capture free) substitution of F for (free occurrences of) X in E. More generally, for a finite set of variables $\{X_1, \ldots, X_n\}$, we write $E\{E_1, \ldots, E_n/X_1, \ldots, X_n\}$ or $E\{E_i/X_i \mid i = 1, \ldots, n\}$ for the expression obtained by simultaneous (capture free) substitution of F_1 for $X_1, \ldots$, F_n for X_n in E. We write $E \equiv F$ when E, F are syntactically identical.

For an expression $E \in \mathcal{E}$, the set of free variables of E, written $fv(E)$, is defined on the structure of E such that

$$fv(\mathbf{0}) = \emptyset, \quad fv(X) = \{X\}, \quad fv(a.E) = fv(E),$$
$$fv(E + F) = fv(E) \cup fv(F), \quad fv(\mu X.E) = fv(E) - \{X\}.$$

Then it is easy to prove that $fv(E)$ is a finite set for all expression $E \in \mathcal{E}$.

The operational semantics of expressions is given by a transition relation and two binary relations between expressions and variables defined as follows.

▶ **Definition 1.** *The transition relation* $\longrightarrow \subseteq \mathcal{E} \times \mathcal{A}_\tau \times \mathcal{E}$ *is the smallest relation such that (as usual we write* $E \xrightarrow{a} E'$ *for* $(E, a, E') \in \longrightarrow$*):*

1. $a.E \xrightarrow{a} E$;
2. *If* $E_1 \xrightarrow{a} E'$ *then* $E_1 + E_2 \xrightarrow{a} E'$;
3. *If* $E_2 \xrightarrow{a} E'$ *then* $E_1 + E_2 \xrightarrow{a} E'$;
4. *If* $E\{\mu X.E/X\} \xrightarrow{a} E'$ *then* $\mu X.E \xrightarrow{a} E'$.

Define $\rhd \subseteq \mathcal{E} \times \mathcal{V}$ *as the smallest relation such that*

5. $X \rhd X$;
6. *If* $E_1 \rhd X$ *then* $E_1 + E_2 \rhd X$;
7. *If* $E_2 \rhd X$ *then* $E_1 + E_2 \rhd X$;
8. *If* $E\{\mu Y.E/Y\} \rhd X$ *then* $\mu Y.E \rhd X$.

Intuitively, $E \rhd X$ means that X has an occurrence in E which is not proceeded by any action, not even a τ.

We follow the CCS (Milner's Calculus of Communicating Systems [6]) tradition to write $\Longrightarrow$ for $(\xrightarrow{\tau})^*$, i.e. the reflexive and transitive closure of $\xrightarrow{\tau}$. We also freely write $\Longrightarrow\xrightarrow{a}$ and $\Longrightarrow\xrightarrow{a}\Longrightarrow$ etc. for various composition of transition relations. We write $E \tilde{\rhd} X$ if $E \Longrightarrow E'$ for some E' with $E' \rhd X$.

▶ **Definition 2.** *A free occurrence of a variable* X *in an expression* $E \in \mathcal{E}$ *is* guarded *if it occurs in a subexpression of the form* $a.F$ *where* $a \neq \tau$. X *is* (un)guarded *in* E *if (not) every free occurrence of* X *in* E *is guarded. An expression* $E \in \mathcal{E}$ *is* guarded *if for every subexpression* $\mu X.F$, X *is guarded in* F.

▶ **Lemma 3.** *Let* $E \in \mathcal{E}, X \in \mathcal{V}$. X *is unguarded in* E *iff* $E \tilde{\rhd} X$.

Proof. Straightforward. ◀

▶ **Theorem 4.** *For* $E \in \mathcal{E}$, *let* $\mathcal{E}_E$ *be the set of expressions which are reachable from* E, *i.e.* $\mathcal{E}_E$ *is the smallest subset of* $\mathcal{E}$ *such that* $E \in \mathcal{E}_E$ *and* $\mathcal{E}_E$ *is closed for transitions (if* $F \xrightarrow{a} G$ *with* $F \in \mathcal{E}_E$ *and* $a \in \mathcal{A}_\tau$ *then* $G \in \mathcal{E}_E$*). Then* $\mathcal{E}_E$ *is a finite set.*

Proof. It was proved in [8] (Proposition 1). ◀

▶ **Definition 5.** *Let* $R \subseteq \mathcal{E} \times \mathcal{E}$ *be a symmetric binary relation between expressions. If for all* $(E, F) \in R$ *the following hold:*

1. *whenever* $E \xrightarrow{a} E'$, *then*
   a. *either* $a = \tau$ *and* $(E', F) \in R$,
   b. *or there exist* F_1, F_2, F' *such that* $F \Longrightarrow F_1, F_1 \xrightarrow{a} F_2, F_2 \Longrightarrow F'$, *and* $(E, F_1), (E', F_2), (E', F') \in R$;
2. *whenever* $E \rhd X$, *then there exists* F' *such that* $F \Longrightarrow F'$, $F' \rhd X$, *and* $(E, F') \in R$;

then R *is called a branching bisimulation.*

If for all $(E, F) \in R$ *the above hold except that without requiring* $(E', F_2) \in R$ *in 1. then* R *is called an* η*-bisimulation.*

If for all $(E, F) \in R$ *the above hold except that without requiring* $(E, F_1) \in R$ *in 1. and* $(E, F') \in R$ *in 2., then* R *is called a delay bisimulation.*

If for all $(E, F) \in R$ *the above hold except that without requiring* $(E, F_1), (E', F_2) \in R$ *in 1. and* $(E, F') \in R$ *in 2., then* R *is called a weak bisimulation.*

Define branching bisimilarity, delay bisimilarity, η*-bisimilarity, weak bisimilarity, written* $\approx_b, \approx_d, \approx_\eta, \approx_w$ *respectively as follows*

$$\approx_b = \bigcup\{R \mid R \textit{ is a branching bisimulation}\}, \quad \approx_d = \bigcup\{R \mid R \textit{ is a delay bisimulation}\},$$

$$\approx_\eta = \bigcup\{R \mid R \textit{ is an } \eta\textit{-bisimulation}\}, \quad \approx_w = \bigcup\{R \mid R \textit{ is a weak bisimulation}\}.$$

▶ Remark. Normally the definition of branching bisimulation does not require the existence of F' in 1., that is because with the existence of F_2 the requirement of F' is trivially satisfied by taking F_2 as F'. For this reason the variations of definition result in the same relation.

With the above definition, it is well-known that $\approx_b, \approx_d, \approx_\eta, \approx_w$ are all equivalence relations. Moreover, it is standard to prove that each of the equivalences is the largest one amongst the corresponding bisimulation relations. Also, about the distinguishing power of these equivalences we have $\approx_b \subseteq \approx_d \subseteq \approx_w$, and $\approx_b \subseteq \approx_\eta \subseteq \approx_w$.

None of the four equivalences is a congruence on $\mathcal{E}$. As a classical counterexample, note that $a.0 \approx_w \tau.a.0$ while $a.0 + b.0 \not\approx_w \tau.a.0 + b.0$, where a, b are different non-τ actions. A standard way to get around this problem, as noted in [9], is to introduced a rootedness condition on top of these equivalences to obtain congruence relations for expressions.

▶ **Definition 6.** *Two expressions E and F are rooted branching bisimilar, notation $E =_b F$, if the following hold:*

1. *whenever $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ such that $E' \approx_b F'$;*
2. *whenever $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ such that $E' \approx_b F'$;*
3. *$E \triangleright X$ if and only if $F \triangleright X$.*

Two expressions E and F are rooted delay bisimilar, notation $E =_d F$, if the following hold:

1. *whenever $E \xrightarrow{a} E'$ then $F \Longrightarrow\xrightarrow{a} F'$ such that $E' \approx_d F'$;*
2. *whenever $F \xrightarrow{a} F'$ then $E \Longrightarrow\xrightarrow{a} E'$ such that $E' \approx_d F'$;*
3. *if $E \triangleright X$ then $F \tilde{\triangleright} X$;*
4. *if $F \triangleright X$ then $E \tilde{\triangleright} X$.*

Two expressions E and F are rooted η-bisimilar, notation $E =_\eta F$, if the following hold:

1. *whenever $E \xrightarrow{a} E'$ then $F \xrightarrow{a}\Longrightarrow F'$ such that $E' \approx_\eta F'$;*
2. *whenever $F \xrightarrow{a} F'$ then $E \xrightarrow{a}\Longrightarrow E'$ such that $E' \approx_\eta F'$;*
3. *$E \triangleright X$ if and only if $F \triangleright X$.*

Two expressions E and F are rooted weak bisimilar, notation $E =_w F$, if the following hold:

1. *whenever $E \xrightarrow{a} E'$ then $F \Longrightarrow\xrightarrow{a}\Longrightarrow F'$ such that $E' \approx_w F'$;*
2. *whenever $F \xrightarrow{a} F'$ then $E \Longrightarrow\xrightarrow{a}\Longrightarrow E'$ such that $E' \approx_w F'$;*
3. *if $E \triangleright X$ then $F \tilde{\triangleright} X$;*
4. *if $F \triangleright X$ then $E \tilde{\triangleright} X$.*

Thus defined, it is easy to prove that the four rooted relations are all congruence relations on the constructions of expressions, and that they are in fact the weakest congruences included in the respective equivalences. From now on we will call $=_b$ branching congruence, $=_d$ delay congruence, $=_\eta$ η-congruence, and $=_w$ weak congruence.

3 Axiomatisation Hierarchy

In [8] van Glabbeek presented an inference system for branching congruence $=_b$. The following is the set of axioms and rules of the inference system (with slight simplification), besides the rules for equational reasoning (reflexivity, symmetry, transitivity, and substituting equal for equal):

S1	$E + F = F + E$	
S2	$E + (F + G) = (E + F) + G$	
S3	$E + E = E$	
S4	$E + \mathbf{0} = E$	
B	$a.(\tau.(E + F) + F) = a.(E + F)$	
R1	$\mu X.E = E\{\mu X.E/X\}$	
R2	if $F = E\{F/X\}$ then $F = \mu X.E$	provided X is guarded in E
R3	$\mu X.(X + E) = \mu X.E$	
R4	$\mu X.(\tau.(\tau.E + F) + G) = \mu X.(\tau.(E + F) + G)$	provided X is unguarded in E
R5	$\mu X.(\tau.(X + E) + F) = \mu X.(\tau.(E + F) + F)$	

We write **SBR** $\vdash E = F$ if $E = F$ can be inferred using the above axioms and rules through equational reasoning, where **SBR** stands for the axioms S1-S4 plus axiom B plus rule and axioms R1-R5. Often we will omit **SBR** and just write $\vdash E = F$.

To obtain a complete axiomatisation for $=_d$, one only needs to add the following axiom T2 into the inference system **SBR**:

T2 $\quad \tau.E + E = \tau.E.$

We call the resulting system **SBRT2**, and write **T2** $\vdash E = F$ if $E = F$ can be inferred in **SBRT2**.

To obtain a complete axiomatisation for $=_\eta$, one only needs to add the following axiom T3 into the inference system **SBR**:

T3 $\quad a.(E + \tau.F) + a.F = a.(E + \tau.F).$

We call the resulting system **SBRT3**, and write **T3** $\vdash E = F$ if $E = F$ can be inferred in **SBRT3**.

Now to obtain a complete axiomatisation for $=_w$, we can add both T2 and T3 into **SBR**. We call the resulting inference system **SBRT2T3**, and write **T2T3** $\vdash E = F$ if $E = F$ can be inferred in **SBRT2T3**.

The name T2 and T3 are from the famous three τ-laws by Hennessy and Milner [3], where they proposed T2 and T3 together with T1: $a.\tau.E = a.E$, to make a complete axiomatisation with respect to $=_w$ for the set of CCS expressions without recursion. In **SBR**, T1 can be easily inferred from B and S4 ($\vdash a.\tau.E = a.(\tau.(E + \mathbf{0}) + \mathbf{0}) = a.(E + \mathbf{0}) = a.E$). Thus T1 is relegated to a theorem in **SBR** as stated in the following lemma, and can be spared from the axioms.

▶ **Lemma 7.** *Let $E \in \mathcal{E}$, then $\vdash a.\tau.E = a.E$.*

The following theorem states the soundness of the inference systems with respect to corresponding congruences.

▶ **Theorem 8.** *Let $E, F \in \mathcal{E}$.*
1. *if $\vdash E = F$ then $E =_b F$;*
2. *if **T2** $\vdash E = F$ then $E =_d F$;*
3. *if **T3** $\vdash E = F$ then $E =_\eta F$;*
4. *if **T2T3** $\vdash E = F$ then $E =_w F$.*

A standard strategy of proving soundness of an axiomatisation with respect to some congruence is first to prove that all the axioms are sound and then to prove that all the inference rules preserve soundness. In this paper, since we concentrate on completeness, we will skip the proof of this theorem. A detailed proof of soundness of **SBR** with respect to $=_b$ can be found in [8] (Corollary 1).

Because the axioms and rules of **SBR** are also axioms and rules of **SBRT2** and of **SBRT3** and of **SBRT2T3**, the following are relationships about these axiomatisations obviously hold.

▶ **Theorem 9.** *Let $E, F \in \mathcal{E}$. If $\vdash E = F$ then **T2** $\vdash E = F$ and **T3** $\vdash E = F$ and **T2T3** $\vdash E = F$.*

The original axioms of the inference system for $=_b$ in [8] also include the equation $\mu X.(\tau.(X+E)+\tau.(X+F)+G) = \mu X.(\tau.(X+E+F)+G)$. However it turns out that this equation can be inferred from S1-S3 and R4-R5, thus it is a derived rule and can be omitted from the set of axioms.

▶ **Definition 10.** *A guarded recursion is an expression of the form $\mu X.E$ where X is guarded in E. An expression is said guarded if every recursive subexpression in it is a guarded recursion.*

▶ **Theorem 11.** *Let $E \in \mathcal{E}$. Then there is a guarded expression E' such that $\vdash E = E'$.*

Proof. Has been proved by van Glabbeek in [8]. ◀

▶ **Lemma 12.** *Let $E \in \mathcal{E}$, E guarded.*
1. *If $E' \in \mathcal{E}_E$ (the transition closure of E), then E' is also guarded.*
2. *There is no infinite τ-transition sequence starting from E ($\xrightarrow{\tau}$ is well-founded).*

Proof. See the proof of Lemma 2 in [8]. ◀

▶ **Lemma 13.** *Let $E \in \mathcal{E}$. Then $\{a.E' \mid E \Longrightarrow\xrightarrow{a}\Longrightarrow E'\}$ and $\{W \mid E \tilde{\triangleright} W\}$ are finite sets.*

Proof. It is easy to prove by induction on the rules which defines $\triangleright$ and $\longrightarrow$, that if $E \triangleright W$ then $W \in fv(E)$, and if $E \xrightarrow{a} E'$ then $fv(E') \subseteq fv(E)$. Thus if $E \tilde{\triangleright} X$ i.e. there is E' such that $E \Longrightarrow E', E' \triangleright X$, then $X \in fv(E') \subseteq fv(E)$. So $\{W \mid E \tilde{\triangleright} W\} \subseteq fv(E)$. Since $fv(E)$ is a finite set, so is $\{W \mid E \tilde{\triangleright} W\}$.

For $E \in \mathcal{E}$, define $sort(E)$ inductively on the structure of E such that:

$$sort(\mathbf{0}) = sort(X) = \emptyset, \qquad sort(E+F) = sort(E) \cup sort(F),$$
$$sort(a.E) = sort(E) \cup \{a\}, \qquad sort(\mu X.E) = sort(E).$$

Then it is easy to prove by induction on the structure of E that $sort(E)$ is a finite set. Next, we prove by induction on the rules which defines the transition relation that if $E \xrightarrow{a} E'$ then $a \in sort(E)$ and $sort(E') \subseteq sort(E)$. Thus it follows that if $E \Longrightarrow\xrightarrow{a}\Longrightarrow E'$ then $a \in sort(E)$. Now to prove that $\{a.E' \mid E \Longrightarrow\xrightarrow{a}\Longrightarrow E'\}$ is a finite set, we note that $\{a.E' \mid E \Longrightarrow\overset{a}{\Longrightarrow}\longrightarrow E'\} \subseteq \{a.E' \mid a \in sort(E), E' \in \mathcal{E}_E\}$. Since $sort(E)$ and $\mathcal{E}_E$ are finite sets, so is $\{a.E' \mid a \in sort(E), E' \in \mathcal{E}_E\}$. Thus $\{a.E' \mid E \Longrightarrow\xrightarrow{a}\Longrightarrow E'\}$ is a finite set. ◀

For a finite set $S = \{E_1, \ldots, E_n\}$ of expressions, let ΣS be an abbreviation for $E_1 + \ldots + E_n$. This notation is justified by the axioms S1-S4. Then with Lemma 13, the Σ notation used in the following lemma is meaningful.

▶ **Lemma 14.** *Let $E \in \mathcal{E}$. Then $\vdash E = \Sigma\{a.E' \mid E \xrightarrow{a} E'\} + \Sigma\{W \mid E \triangleright W\}$.*

Proof. It was proved in [8] (Lemma 6). ◀

▶ **Lemma 15.** *Let $E \in \mathcal{E}$. Then*
1. *if $E \xrightarrow{a} E'$ then $\vdash E = E + a.E'$;*
2. *if $E \rhd X$ then $\vdash E = E + X$;*
3. *if $E \Longrightarrow\xrightarrow{a} E'$ then $\textbf{\textit{T2}} \vdash E = E + a.E'$;*
4. *if $E \tilde{\rhd} W$, then $\textbf{\textit{T2}} \vdash E = E + W$;*
5. *if $E \xrightarrow{a}\Longrightarrow E'$ then $\textbf{\textit{T3}} \vdash E = E + a.E'$;*
6. *if $E \Longrightarrow\xrightarrow{a}\Longrightarrow E'$ then $\textbf{\textit{T2T3}} \vdash E = E + a.E'$.*

Proof. 1. and 2. immediately follows from Lemma 14 and S3.

The rest can be proved by induction on the length of the τ-transition sequence in $\Longrightarrow$. Here we prove 3. as follows. Suppose $E \Longrightarrow\xrightarrow{a} E'$. If the length of the τ-transition sequence in $\Longrightarrow$ is 0, then in this case $E \xrightarrow{a} E'$, which is *1*. If the length of the τ-transition sequence in $\Longrightarrow$ is greater than 0, then there is E'' such that $E \Longrightarrow\xrightarrow{\tau} E''$, $E'' \xrightarrow{a} E'$, and by the induction hypothesis $\mathbf{T2} \vdash E = E + \tau.E''$ and $\mathbf{T2} \vdash E'' = E'' + a.E'$. Now we have the following reasoning in **SBRT2**:

$$\begin{aligned}
\mathbf{T2} \vdash E \; &= E + \tau.E'' && \text{(IH)}\\
&= E + \tau.(E'' + a.E') && \text{(IH)}\\
&= E + \tau.(E'' + a.E') + E'' + a.E' && \text{(T2)}\\
&= E + \tau.(E'' + a.E') + E'' + a.E' + a.E' && \text{(S3)}\\
&= E + \tau.(E'' + a.E') + a.E' && \text{(T2)}\\
&= E + \tau.E'' + a.E'\\
&= E + a.E'
\end{aligned}$$
◀

▶ **Lemma 16.** *Let $E \in \mathcal{E}$. Then*
1. *$\textbf{\textit{T2}} \vdash E = \Sigma\{a.E' \mid E \Longrightarrow\xrightarrow{a} E'\} + \Sigma\{W \mid E \tilde{\rhd} W\}$;*
2. *$\textbf{\textit{T3}} \vdash E = \Sigma\{a.E' \mid E \xrightarrow{a}\Longrightarrow E'\} + \Sigma\{W \mid E \rhd W\}$;*
3. *$\textbf{\textit{T2T3}} \vdash E = \Sigma\{a.E' \mid E \Longrightarrow\xrightarrow{a}\Longrightarrow E'\} + \Sigma\{W \mid E \tilde{\rhd} W\}$.*

Proof. For 1. we have the following reasoning in **SBRT2**:

$$\begin{aligned}
\mathbf{T2} \vdash E \; &= E + \Sigma\{a.E' \mid E \Longrightarrow\xrightarrow{a} E'\} + \Sigma\{W \mid E \tilde{\rhd} W\} && \text{(3. and 4. of Lemma 15)}\\
&= \Sigma\{a.E' \mid E \xrightarrow{a} E'\} + \Sigma\{W \mid E \rhd W\}\\
&\quad + \Sigma\{a.E' \mid E \Longrightarrow\xrightarrow{a} E'\} + \Sigma\{W \mid E \tilde{\rhd} W\} && \text{(Lemma 14)}\\
&= \Sigma\{a.E' \mid E \Longrightarrow\xrightarrow{a} E'\} + \Sigma\{W \mid E \tilde{\rhd} W\} && \text{(S3)}
\end{aligned}$$

2. and 3. can be proved as 1. by using Lemma 15 and Lemma 14. ◀

4 Recursive Specifications and Provability

We now describe the recursive equations mentioned in the introduction. They are formally called recursive specifications.

▶ **Definition 17.** *A* recursive specification *S is a finite set of equations*

$$\{X_i = F_i \mid i = 0, 1, \ldots, n-1\}$$

where n different variables $X_0, \ldots, X_{n-1}$ are called the formal variables *of S, and $F_i \in \mathcal{E}$ for $i = 0, \ldots, n-1$. For $E \in \mathcal{E}$, E is said to* T-provably solve (or satisfy) *the recursive specification S above in the variable $X_k \in V_S$ if there are expressions E_i for $i = 0, \ldots, n-1$*

with E being E_k, such that $T \vdash E_i = F_i\{E_j/X_j \mid j = 0, \ldots, n-1\}$ holds for $i = 0, \ldots, n-1$. Define a relation $\xrightarrow{u}_S$ between the formal variables of S such that $X_i \xrightarrow{u}_S X_j$ if $F_i \tilde{\rhd} X_j$. S is said to be guarded *if there is no infinite $\xrightarrow{u}_S$ transition sequence starting from any $X_i \in V_S$.*

▶ **Theorem 18.** *(Unique solution) If S is a recursive specification with formal variable X_0, then there is an expression E which* ***SBR****-provably solves S in X_0. Moreover if S is guarded and there are two such expressions E and F which both* ***SBR****-provably solve S in X_0, then $\vdash E = F$.*

Proof. In Milner [7], Theorem 4.2. ◀

▶ **Definition 19.** *Let $\mathcal{E}_0 \subseteq \mathcal{E}$. $\mathcal{E}_0$ is called a* simple set *if*

1. *$\mathcal{E}_0$ is a finite set;*
2. *$\mathcal{E}_0$ is transition closed, i.e. whenever $E \in \mathcal{E}_0$ and $E \xrightarrow{a} E'$ then $E' \in \mathcal{E}_0$;*
3. *$\xrightarrow{\tau}$ is well-founded in $\mathcal{E}_0$, i.e. there does not exist an infinite sequence of τ-transitions starting from any element in $\mathcal{E}_0$.*

The key step in our completeness proof is to show that if E, F are two expressions in the same simple set $\mathcal{E}_0$ such that $E \approx_b F$, then $\vdash \tau.E = \tau.F$ (in [2] Deng proved this *promotion lemma* for finite processes, which plays a key role in his completeness results). With this we go on to prove the completeness of **SBR** with respect to $=_b$. Since the detailed construction is quite technical, before starting we spend a few words to explain the intuition behind it. For an equivalence relation, let us say $\approx_b$, one can quotient $\mathcal{E}_0$ into equivalence classes, and then construct a minimal labeled transition system such that each equivalence class as a state of the constructed transition system mimics the behaviour of its members. If we can write expressions to describe the constructed labeled transition system, then there is a good chance that we can use the expressions to solve a recursive specification.

In the rest of this section, we will fix a simple set $\mathcal{E}_0$ and construct recursive specifications with respect to it.

Since $\mathcal{E}_0$ is a finite set, the equivalence relation $\approx_b$ partitions it into a finite number of equivalence classes. Thus, we can assume that $\approx_b$ partitions $\mathcal{E}_0$ into n $\approx_b$-equivalence classes, and we write $\{C_1, \ldots, C_n\}$ for the partition. For $E \in C_i$, E is called a *bottom element of* C_i if whenever $E \xrightarrow{\tau} E'$ then $E' \notin C_i$. That is to say, a bottom element of an equivalence class cannot remain in the same class after performing a τ-transition. Take any $E \in C_i$, if E is not a bottom element of C_i, then we can find $E' \in C_i$ such that $E \xrightarrow{\tau} E'$, and since $\xrightarrow{\tau}$ is well-founded for elements of $\mathcal{E}_0$, this cannot go on for ever, which implies that bottom elements must exist in C_i. Now let us fix n expressions $B_1, \ldots, B_n$ such that each B_i is a bottom element of C_i. For $E \in \mathcal{E}_0$, if $E \in C_i$ we call i the index of E, and we write $\iota(E)$ for the index of E. With this notation, the following are two obvious relations hold for the expressions in $\mathcal{E}_0$: A) for all $E \in \mathcal{E}_0$, $E \approx_b B_{\iota(E)}$; B) for all $E, F \in \mathcal{E}_0$, $E \approx_b F$ if and only if $\iota(E) = \iota(F)$.

▶ **Definition 20.** *Define a recursive specification $R^{\mathcal{E}_0} = \{Z_i = H_i \mid i = 1, \ldots, n\}$, where $Z_1, \ldots, Z_n \in \mathcal{V}$ are n different variables which do not occur free in any expressions in $\mathcal{E}_0$, and each H_i is defined as follows:*

$$H_i \equiv \Sigma\{a.Z_{\iota(E)} \mid B_i \xrightarrow{a} E\} + \Sigma\{W \mid B_i \rhd W\}.$$

Now with the recursive specification $R^{\mathcal{E}_0}$ defined above, according to Theorem 18, there exist n expressions $D_1, \dots, D_n$, which **SBR**-provably satisfy $R^{\mathcal{E}_0}$, i.e. the following holds for $i = 1, \dots, n$:

$$\vdash D_i = H_i\{D_1, \dots, D_n/Z_1, \dots, Z_n\}. \qquad (1)$$

In fact these D's can be constructed so that they form a canonical solution to $R^{\mathcal{E}_0}$ in that the transition space consists of only $D_1, \dots, D_n$, which is minimal in size since $D_i \not\approx_b D_j$ when $i \neq j$. However as we only need equality (1) without referring to the actual behaviour of the D's, here we will not further argue the canonicity of the D's (the canonicity and other properties of D's will be clear after Theorem 29 in the next section). Next we will concentrate on using the D's to solve some useful recursive specifications.

▶ **Lemma 21.** *Let $E \in \mathcal{E}_0$. If E is a bottom element, then*

$$\vdash D_{\iota(E)} = \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \triangleright W\}.$$

Proof. Since $\vdash D_{\iota(E)} = H_{\iota(E)}\{D_i/Z_i \,|\, i = 1, \dots, n\}$ (equality (1) above), to prove the lemma, we only need to establish:

$$\vdash H_{\iota(E)}\{D_i/Z_i \mid i = 1, \dots, n\} = \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \triangleright W\}.$$

According to Definition 20, the left hand side of the above equation is $\Sigma\{a.D_{\iota(F)} \mid B_{\iota(E)} \xrightarrow{a} F\} + \Sigma\{W \mid B_{\iota(E)} \triangleright W\}$ where $B_{\iota(E)}$ is the chosen bottom element in the equivalence class of E. As both E and $B_{\iota(E)}$ are bottom elements of the same equivalence class, a simple fact is that $E \triangleright W$ if and only if $B_{\iota(E)} \triangleright W$, and $E \xrightarrow{a} F$ if and only if $B_{\iota(E)} \xrightarrow{a} F'$ such that $\iota(F) = \iota(F')$ (this easily follows from Definition 5 and the condition that bottom elements cannot perform τ-transition without moving state to a different equivalence class). From this simple fact it follows that

$$\vdash \Sigma\{a.D_{\iota(F)} \,|\, B_{\iota(E)} \xrightarrow{a} F\} + \Sigma\{W \,|\, B_{\iota(E)} \triangleright W\} = \Sigma\{a.D_{\iota(F)} \,|\, E \xrightarrow{a} F\} + \Sigma\{W \,|\, E \triangleright W\}$$

as both sides has the same set of summands. ◀

▶ **Lemma 22.** *Let $E \in \mathcal{E}_0$, then*
$\vdash \tau.D_{\iota(E)} + D_{\iota(E)} = \tau.D_{\iota(E)} + D_{\iota(E)} + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \triangleright W\}$.

Proof. Since $\vdash D_{\iota(E)} = H_{\iota(E)}\{D_i/Z_i \,|\, i = 1, \dots, n\}$ (equality (1) above), to prove the lemma we only need to establish:

$$\begin{aligned} \vdash \quad & \tau.D_{\iota(E)} + H_{\iota(E)}\{D_i/Z_i \mid i = 1, \dots, n\} = \\ & \tau.D_{\iota(E)} + H_{\iota(E)}\{D_i/Z_i \mid i = 1, \dots, n\} + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \triangleright W\}. \end{aligned}$$

For that, with axioms S1, S2, S3, we need to show that the extra summands on the right hand side of the equality in $\Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \triangleright W\}$ are also summands in $\tau.D_{\iota(E)} + H_{\iota(E)}\{D_i/Z_i \mid i = 1, \dots, n\}$, which can be verified with the condition that $E \approx_b B_{\iota(E)}$ and $B_{\iota(E)}$ is a bottom element of $C_{\iota(E)}$. Consider the summand W with $E \triangleright W$. Since $E \approx_b B_{\iota(E)}$, there exists $E' \in C_{\iota(E)}$ such that $B_{\iota(E)} \Longrightarrow E', E' \triangleright W$. Because $B_{\iota(E)}$ is a bottom element which cannot perform any τ while staying within $C_{\iota(E)}$, E' must be $B_{\iota(E)}$, thus according to the definition of H_i in Definition 20 W is a summand in $H_{\iota(E)}$, and also in $H_{\iota(E)}\{D_i/Z_i \mid i = 1, \dots, n\}$. Consider the summand $a.D_{\iota(F)}$ with $E \xrightarrow{a} F$. Since $E \approx_b B_{\iota(E)}$, then either $a = \tau$ and $F \in B_{\iota(E)}$ and in this case $a.D_{\iota(F)}$ is just $\tau.D_{\iota(E)}$, or there exists $E' \in B_{\iota(E)}$ such that $B_{\iota(E)} \Longrightarrow E', E' \xrightarrow{a} F'$ such that $F \approx_b F'$. Because $B_{\iota(E)}$ is a bottom element, E' must be $B_{\iota(E)}$, according to the definition of H_i in Definition 20, $a.Z_{\iota(F')}$ is a summand in $H_{\iota(E)}$, thus $a.D_{\iota(F')}$ (which in this case is the same as $a.D_{\iota(F)}$) is a summand in $H_{\iota(E)}\{D_i/Z_i \mid i = 1, \ldots, n\}$. ◀

▶ **Lemma 23.** *Let $E \in \mathcal{E}_0$, then $\vdash \tau.D_{\iota(E)} = \tau.(\Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\})$.*

Proof. If E is a bottom element, then the lemma follows from Lemma 21.

If E is not a bottom element, i.e. there is E' such that $E \xrightarrow{\tau} E'$ and $\iota(E) = \iota(E')$, then

$$\begin{aligned}
&\vdash \tau.D_{\iota(E)} = \tau.(\tau.D_{\iota(E)} + D_{\iota(E)}) && \text{(B)}\\
&= \tau.(\tau.D_{\iota(E)} + D_{\iota(E)} + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(Lemma 22)}\\
&= \tau.(\tau.(\tau.D_{\iota(E)} + D_{\iota(E)} + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\})\\
&\quad + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(B)}\\
&= \tau.(\tau.(\tau.D_{\iota(E)} + D_{\iota(E)}) + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(Lemma 22)}\\
&= \tau.(\tau.D_{\iota(E)} + \Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(B)}\\
&= \tau.(\Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}). && (\dagger)
\end{aligned}$$

$\dagger$ is because $\iota(E) = \iota(E')$ and $\tau.D_{\iota(E)}$ is the same as $\tau.D_{\iota(E')}$, while $\tau.D_{\iota(E')}$ is a summand in $\Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\}$. ◀

▶ **Lemma 24** (Promotion)**.** *Let $E_1, E_2 \in \mathcal{E}_0$. If $E_1 \approx_b E_2$ then $\vdash \tau.E_1 = \tau.E_2$.*

Proof. First note that for each $E \in \mathcal{E}_0$ the following holds:

$$\begin{aligned}
&\vdash \tau.D_{\iota(E)} = \tau.(\Sigma\{a.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(Lemma 23)}\\
&= \tau.(\Sigma\{a.\tau.D_{\iota(F)} \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}). && \text{(Lemma 7)}
\end{aligned}$$

On the other hand

$$\begin{aligned}
&\vdash \tau.E = \tau.(\Sigma\{a.F \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}) && \text{(Lemma 14)}\\
&= \tau.(\Sigma\{a.\tau.F \mid E \xrightarrow{a} F\} + \Sigma\{W \mid E \rhd W\}). && \text{(Lemma 7)}
\end{aligned}$$

Thus, for each $E \in \mathcal{E}_0$ it holds that $\vdash \tau.D_{\iota(E)} = \tau.E$, because both $\tau.D_{\iota(E)}$ and $\tau.E$ **SBR**-provably solve the following guarded recursive specification on X_E:

$$S = \{X_E = \tau.(\Sigma\{a.X_F \mid E \xrightarrow{a} F\}) + \Sigma\{W \mid E \rhd W\} \mid E \in \mathcal{E}_0\}.$$

The well-foundedness of $\xrightarrow{u}_S$ easily follows from the well-foundedness of $\xrightarrow{\tau}$ in $\mathcal{E}_0$ which is a simple set. Thus S is a guarded recursive specification. Now if $E_1 \approx_b E_2$, then $\iota(E_1) = \iota(E_2)$. Thus $\vdash \tau.E_1 = \tau.D_{\iota(E_1)} = \tau.D_{\iota(E_2)} = \tau.E_2$. ◀

▶ **Theorem 25** (Completeness of **SBR**)**.** *Let $E, F \in \mathcal{E}$. If $E =_b F$ then $\vdash E = F$.*

Proof. By Lemma 11 there exist guarded expressions E',F' such that $\vdash E = E'$, $\vdash F = F'$. By the soundness of **SBR**, $E' =_b E =_b F =_b F'$. Let $\mathcal{E}_0 = \mathcal{E}_{E'} \cup \mathcal{E}_{F'}$, where $\mathcal{E}_{E'}, \mathcal{E}_{F'}$ are the sets of expressions reachable from E' and F' respectively. $\mathcal{E}_0$ is obviously transition closed. By Theorem 4 $\mathcal{E}_0$ is a finite set. By Lemma 12 $\xrightarrow{\tau}$ is well-founded in $\mathcal{E}_0$. Thus $\mathcal{E}_0$ is a simple set. By Lemma 14 $\vdash F' = \Sigma S_1 + \Sigma S_2$ where $S_1 = \{a.F'' \mid F' \xrightarrow{a} F''\}$, $S_2 = \{W \mid F' \rhd W\}$. According to Lemma 13 S_1, S_2 are finite sets, let $S_1 = \{a_1.F_1, \ldots, a_k.F_k\}$, $S_2 = \{W_1, \ldots, W_m\}$.

Because $E' =_b F'$, for each $a_i.F_i \in S_1$ there is E'' such that $E' \xrightarrow{a_i} E''$ and $E'' \approx_d F_i$, then by Lemma 7 and Lemma 24 (note that $E'', F_i \in \mathcal{E}_0$) $\vdash a_i.E'' = a_i.\tau.E'' = a_i.\tau.F_i = a_i.F_i$, and by Lemma 15 $\vdash E' = E' + a_i.E''$. It follows that $\vdash E' = E' + a_i.F_i (A)$. For each $W_j \in S_2$, since $E' =_b F'$, $E' \rhd W_j$, then by Lemma 15 $\vdash E' = E' + W_j (B)$. After these preparation we have the following reasoning in **SBR**:

$$\begin{array}{rcll}
\vdash E' + F' & = & E' + \Sigma_{i=1}^{k} a_i.F_i + \Sigma_{j=1}^{m} W_j & \text{(Lemma 14)} \\
& = & E' + a_1.F_1 + \Sigma_{i=2}^{k} a_i.F_i + \Sigma_{j=1}^{m} W_j & \\
& = & E' + \Sigma_{i=2}^{k} a_i.F_i + \Sigma_{j=1}^{m} W_j & \text{(A above)} \\
& = & E' + \Sigma_{j=1}^{m} W_j & \text{(after } k \text{ steps)} \\
& = & E' + W_1 + \Sigma_{j=2}^{m} W_j & \\
& = & E' + \Sigma_{j=2}^{m} W_j & \text{(B above)} \\
& = & E' & \text{(after } m \text{ steps)}
\end{array}$$

In the same way we can prove $\vdash E' + F' = F'$. Thus $\vdash E' = F'$, and $\vdash E = F$. ◀

To summarize, so far we proved the completeness of **SBR** with respect to $=_b$ by proving promotion lemma (Lemma 24) through constructing solution to the recursive specification $R^{\mathcal{E}_0}$. This plan of proof can be easily adapted to work for the completeness of **SBRT2** w.r.t $=_d$, that of **SBRT3** w.r.t $=_\eta$, and that of **SBRT2T3** w.r.t. $=_w$. Very few adjustment is required, including to quotient with proper equivalence (of course), to change the transition $B_i \xrightarrow{a} E$ in $R^{\mathcal{E}_0}$ (Definition 20) accordingly to $B_i \Longrightarrow\xrightarrow{a} E$ and $B_i \xrightarrow{a}\Longrightarrow E$ and $B_i \Longrightarrow\xrightarrow{a}\Longrightarrow E$, and in establishing the corresponding promotion lemma to use results in Lemma 16 instead of Lemma 14 to work out proper recursive specifications corresponding to the S in Lemma 24. Instead of going through the same plan to establish the other three completeness results, here we shall satisfy ourselves by applying the ideas which are needed in proving the completeness of **SBRT2T3** on deriving equality in **SBRT2T3** of two concrete expressions.

▶ **Example 26.** Let E and F be $b.\mu X.(a.a.X + \tau.\mu Y.a.Y)$ and $b.\mu Z.\tau.a.Z$ respectively, in this example we show **SBRT2T3** $\vdash E = F$. The two expressions are modified from an example in [6] (the modification is to prevent F being able to solve the recursive specification of E easily). By 3. of Lemma 16 the following equalities are provable in **SBRT2T3**:

$$\begin{array}{lrcl}
\textbf{T2T3} \vdash & E & = & b.\mu X.(a.a.X + \tau.\mu Y.a.Y) + b.\mu Y.a.Y \\
\textbf{T2T3} \vdash & \mu X.(a.a.X+\tau.\mu Y.a.Y) & = & a.a.\mu X.(a.a.X+\tau.\mu Y.a.Y) + \tau.\mu Y.a.Y + a.\mu Y.a.Y \\
\textbf{T2T3} \vdash & a.\mu X.(a.a.X+\tau.\mu Y.a.Y) & = & a.\mu X.(a.a.X + \tau.\mu Y.a.Y) + a.\mu Y.a.Y \\
\textbf{T2T3} \vdash & \mu Y.a.Y & = & a.\mu Y.a.Y \\
\textbf{T2T3} \vdash & F & = & b.\mu Z.\tau.a.Z + b.a.\mu Z.\tau.a.Z \\
\textbf{T2T3} \vdash & \mu Z.\tau.a.Z & = & \tau.a.\mu Z.\tau.a.Z + a.\mu Z.\tau.a.Z + a.a.\mu Z.\tau.a.Z \\
\textbf{T2T3} \vdash & a.\mu Z.\tau.a.Z & = & a.\mu Z.\tau.a.Z + a.a.\mu Z.\tau.a.Z
\end{array}$$

So, E and F are **SBRT2T3**-provably solve the following guarded recursive specification S (below on the left) in X_1 and Y_1 respectively, with $E_2 \equiv \mu X.(a.a.X + \tau.\mu Y.a.Y), E_3 \equiv a.\mu X.(a.a.X + \tau.\mu Y.a.Y), E_4 \equiv \mu Y.a.Y$ provably solving S in X_2, X_3, X_4 respectively, and $F_2 \equiv \mu Z.\tau.a.Z, F_3 \equiv a.\mu Z.\tau.a.Z$ provably solving S in Y_2, Y_3 respectively:

$$
\begin{array}{rrcl|rrcl}
S: & X_1 & = & b.X_2 + b.X_4 & T: & X_1 & = & b.X_2 + b.X_4 \\
 & X_2 & = & a.X_3 + \tau.X_4 + a.X_4 & & X_2 & = & \tau.(a.X_3 + \tau.X_4 + a.X_4) \\
 & X_3 & = & a.X_2 + a.X_4 & & X_3 & = & \tau.(a.X_2 + a.X_4) \\
 & X_4 & = & a.X_4 & & X_4 & = & \tau.a.X_4 \\
 & Y_1 & = & b.Y_2 + b.Y_3 & & Y_1 & = & b.Y_2 + b.Y_3 \\
 & Y_2 & = & \tau.Y_3 + a.Y_2 + a.Y_3 & & Y_2 & = & \tau.(\tau.Y_3 + a.Y_2 + a.Y_3) \\
 & Y_3 & = & a.Y_2 + a.Y_3 & & Y_3 & = & \tau.(a.Y_2 + a.Y_3)
\end{array}
$$

Then by using Lemma 7, we easily obtain that E and F **SBRT2T3**-provably solve the guarded recursive specification T (above on the right) in X_1 and Y_1 respectively, with $\tau.E_2, \tau.E_3, \tau.E_4$ provably solving T in X_2, X_3, X_4, and $\tau.F_2, \tau.F_3$ in Y_2, Y_3 respectively.

On the other hand, the minimal reachable state space containing E and F is $\{E, E_2, E_3, E_4, F, F_2, F_3\}$, which is divided into two $\approx_w$-equivalence classes $C_1 = \{E, F\}$ with bottom element F, and $C_2 = \{E_2, E_3, E_4, F_2, F_3\}$ with bottom element F_3. Using the construction of Definition 20, from C_1 and C_2 we obtain recursive specification $R = \{Z_1 = b.Z_2, Z_2 = a.Z_2\}$, which is provably solved by $b.\mu Z.a.Z$ and $\mu Z.a.Z$ in Z_1 and Z_2 respectively in **SBRT2T3**. Now it is easy to see that $b.\mu Z.a.Z$ provably solve T in both X_1 and Y_1, with $\tau.\mu Z.a.Z$ provably solving T in all X_2, X_3, X_4, Y_2, Y_3. Finally with E and $b.\mu Z.a.Z$ both provably solving T in X_1 and F and $b.\mu Z.a.Z$ both provably solving T in Y_1 where T is guarded, we obtain **T2T3** $\vdash E = b.\mu Z.a.Z$ and **T2T3** $\vdash F = b.\mu Z.a.Z$, hence **T2T3** $\vdash E = F$.

5 Saturation Results and Completeness of All Axiomatisations

With the completeness result of **SBR** with respect to $=_b$, we can also use the method employed in [1] to obtain the completeness results of the extended axiomatisations, i.e. by using the notion of saturation to reduce the completeness of the extended axiomatisations to the established completeness of **SBR** w.r.t $=_b$. With this approach, besides the completeness results, the saturation results (Theorem 30) are interesting in their own rights.

▶ **Definition 27.** *A set of expressions S is called* saturated *if for each $E \in S$:*

1. *whenever $E \xrightarrow{a}\xrightarrow{\tau} F$ then $E \xrightarrow{a} F$;*
2. *whenever $E \xrightarrow{\tau}\xrightarrow{a} F$ then $E \xrightarrow{a} F$;*
3. *whenever $E \xrightarrow{\tau} E'$ and $E' \triangleright X$ then $E \triangleright X$.*

S is called η-saturated if 1. is required to hold for each $E \in S$, and S is called d-saturated if 2. and 3. are required to hold for each $E \in S$.

An expression $E \in \mathcal{E}$ is called saturated *(and* η-saturated, d-saturated *respectively) if there is some $S \subseteq \mathcal{E}$ with $E \in S$ such that S is transition closed and saturated (and η-saturated, d-saturated respectively).*

▶ **Theorem 28.** *Let $E, F \in \mathcal{E}$.*

1. *If E and F are d-saturated, then $E =_d F$ implies $E =_b F$;*
2. *If E and F are η-saturated, then $E =_\eta F$ implies $E =_b F$;*
3. *If E and F are saturated, then $E =_w F$ implies $E =_b F$.*

Proof. See [1], Theorem 4.6. ◀

To prove the main result of this section, Theorem 30, we need the following theorem which is a strengthened version of the existence part in Theorem 18 (a proof can be found in the appendix).

▶ **Theorem 29.** *Let $\{X_i = F_i \,|\, i = 1, \dots, n\}$ be a recursive specification. Then there exist n expressions $E_1, \dots, E_n$ such that E_i **SBR**-provably solves $\{X_i = F_i \,|\, i = 1, \dots, n\}$ in variable X_i. Moreover, the following hold for $i = 1, \dots, n$:*

for $a \in \mathcal{A}_\tau, E \in \mathcal{E}$, $E_i \xrightarrow{a} E$ if and only if $F_i\{E_1, \dots, E_n/X_1, \dots, X_n\} \xrightarrow{a} E$ and for $W \in \mathcal{V}$, $E_i \triangleright W$ if and only if $F_i\{E_1, \dots, E_n/X_1, \dots, X_n\} \triangleright W$.

▶ **Theorem 30** (Saturation). *Let E be a guarded expression. Then there exist guarded expressions E_d, E_η, and E_w, such that E_d is d-saturated and **T2** $\vdash E = E_d$, E_η is η-saturated and **T3** $\vdash E = E_\eta$, and E_w is saturated and **T2T3** $\vdash E = E_w$.*

Proof. Here we only show the existence of E_w. By the same procedure we can construct E_d and E_η. As E is a guarded expression, we can show that $\mathcal{E}_E$ (the transition closure of E) is a simple set (in the same way to show that $\mathcal{E}_0$ is a simple set in the proof of Theorem 25). By Lemma 16 for each $F \in \mathcal{E}_E$ it holds that

$$\mathbf{T2T3} \vdash F = \Sigma\{a.F' \mid F \Longrightarrow\xrightarrow{a}\Longrightarrow F'\} + \Sigma\{W \mid F \tilde{\triangleright} W\}.$$

Thus each $F \in \mathcal{E}_E$ provably solves X_F in the following recursive specification $S^{\mathcal{E}_E}$:

$$\{X_F = \Sigma\{a.X_{F'} \mid F \Longrightarrow\xrightarrow{a}\Longrightarrow F'\} + \Sigma\{W \mid F \tilde{\triangleright} W\} \mid F \in \mathcal{E}_E\}.$$

According to Theorem 29, there exist a set of expressions $\{D_F \,|\, F \in \mathcal{E}_E\}$ such that $D_F \xrightarrow{a} H$ for some expression H if and only if $\Sigma\{a.D_{F'} \,|\, F \Longrightarrow\xrightarrow{a}\Longrightarrow F'\} + \Sigma\{W \,|\, F \tilde{\triangleright} W\} \xrightarrow{a} H$ and $D_F \triangleright W$ for some $W \in \mathcal{V}$ if and only if $\Sigma\{a.D_{F'} \,|\, F \Longrightarrow\xrightarrow{a}\Longrightarrow F'\} + \Sigma\{W \,|\, F \tilde{\triangleright} W\} \triangleright W$. Note that $\Sigma\{a.D_{F'} \,|\, F \Longrightarrow\xrightarrow{a}\Longrightarrow F'\} + \Sigma\{W \,|\, F \tilde{\triangleright} W\} \xrightarrow{a} H$ if and only if $H \equiv D_{F'}$ for some F' such that $F \Longrightarrow\xrightarrow{a}\Longrightarrow F'$, thus $D_F \xrightarrow{a} H$ if and only if $H \equiv D_{F'}$ where $F \Longrightarrow\xrightarrow{a}\Longrightarrow F'$. Which shows that $\{D_F \mid F \in \mathcal{E}_E\}$ is transition closed. Next we show that $\{D_F \mid F \in \mathcal{E}_E\}$ is saturated. For that, take arbitrary $F \in \mathcal{E}_E$, and suppose $D_F \xrightarrow{\tau} H_1 \xrightarrow{a} H_2$. Then there is F_1 such that $F \Longrightarrow\xrightarrow{\tau}\Longrightarrow F_1$ and $H_1 \equiv D_{F_1}$ and also there is F_2 such that $F_1 \Longrightarrow\xrightarrow{a}\Longrightarrow F_2$ and $H_2 \equiv D_{F_2}$, and in this case $F \Longrightarrow\xrightarrow{a}\Longrightarrow F_2$, thus $D_F \xrightarrow{a} D_{F_2} \equiv H_2$. In the same way we can show that if $D_F \xrightarrow{a} H_1 \xrightarrow{\tau} H_2$ then $D_F \xrightarrow{a} H_2$, and if $D_F \xrightarrow{\tau} H, H \triangleright W$ then $D_F \triangleright W$. Thus $\{D_F \mid F \in \mathcal{E}_E\}$ is a saturated set, and D_E is a saturated expression. Since E is guarded, $\xrightarrow{\tau}$ is well-founded in $\mathcal{E}_E$, from which it easily follows that $S^{\mathcal{E}_E}$ is a guarded recursive specification. Now both E and D_E provably solve $S^{\mathcal{E}_E}$ in the variable X_E, and $S^{\mathcal{E}_E}$ is guarded, thus by Theorem 18 **T2T3** $\vdash E = D_E$, and D_E is saturated since it is in the closed and saturated set $\{D_F \mid F \in \mathcal{E}_E\}$. Thus we find D_E for the wanted E_w. ◀

With Theorem 30, the rest of the completeness results follows from the completeness of **SBR** w.r.t. $=_b$.

▶ **Theorem 31.** *Let $E, F \in \mathcal{E}$.*
1. *if $E =_d F$ then **T2** $\vdash E = F$;*
2. *if $E =_\eta F$ then **T3** $\vdash E = F$;*
3. *if $E =_w F$ then **T2T3** $\vdash E = F$.*

Proof. Here we only prove 1., the rest can be established in the same way. Let $E =_d F$. By Theorem 11 there exist guarded expressions E_1, F_1 such that $\vdash E = E_1, \vdash F = F_1$, thus also **T2** $\vdash E = E_1$, **T2** $\vdash F = F_1$. By Theorem 30 there exist d-saturated and guarded expressions

E_2, F_2 such that $\mathbf{T2} \vdash E_1 = E_2, \mathbf{T2} \vdash F_1 = F_2$. By the soundness of **SBRT2** (Theorem 8), $E =_d E_1 =_d E_2$ and $F =_d F_1 =_d F_2$, thus $E_2 =_d F_2$. Since E_2, F_2 are both d-saturated, by Theorem 28, $E_2 =_b F_2$ follows from $E_2 =_d F_2$. Then $\vdash E_2 = F_2$ follows from the completeness of **SBR**, thus also $\mathbf{T2} \vdash E_2 = F_2$, and $\mathbf{T2} \vdash E = E_1 = E_2 = F_2 = F_1 = F$. ◀

6 Related Work

In [5], Milner proposed a set of axioms and rules to infer strong congruence (where τ is treated just as any other action) for regular behaviours and proved the completeness of the inference system by merging the equation sets. That was the first time when the equation set merging strategy was introduced. Later in [7], Milner proposed a set of axioms and rules to infer weak congruence for regular behaviours and used the same strategy to prove the completeness of the inference system. Although our axiomatisation for $=_w$ in this paper is obtained by expanding van Glabbeek's axiomatisation for $=_b$ and has different axioms than Milner's axiomatisation for $=_w$, it can be proved that the two axiomatisations are equivalent in the sense that all the axioms and rules of one can be derived in the other axiomatisation and vise versa.

In [8], van Glabbeek proposed a complete axiomatisation of branching congruence for regular expressions. His axiomatisation is the base for the axiomatisations in the hierarchy studied in this paper. He used Milner's strategy of merging recursive equation sets (recursive specifications) to arrive at the completeness result. In the proof of the main theorem of completeness for guarded expressions, to bridge the gap between equivalence and congruence he used a construction which is an implicit form of the promotion lemma. In principle van Glabbeek's construction can be adapted to work for the completeness proof of the extended axiomatisations, however the interplay between such construction and the inevitable process of saturation could become messy when merging the recursive equation sets.

In [4], Lohrey and the co-authors presented an axiomatisation hierarchy for divergence sensitive weak congruences. They also used the strategy of merging recursive equation sets to arrive at the completeness results. It is expected that our approach could work for divergence sensitive variations of bisimulation based congruences which have not been treated here.

In [2], based on the promotion lemma, Deng presented a uniform completeness proof for the axiomatisations of five congruences: branching congruence, η-congruence, quasi-branching congruence, and weak congruence in the basic CCS without recursion. We have not treated quasi-branching congruence which is sufficiently similar to branching congruence and on which there shall be no difficulty to apply our method.

In [1], Aceto and the co-authors gave complete axiomatisations of branching, delay, weak, and η-congruences for expressions which use prefix iteration instead of recursion to generate infinite behaviours. Prefix iteration is a simpler syntax than recursion in that a normal form exists for every expression of that kind, as a result the complex strategy of joining equation sets is not needed in that case. The work about saturation results in section 6 of this paper follows closely the framework laid out in [1].

7 Conclusion

In this paper we put up a hierarchy of axiomatisations of four well-known congruences with various τ-abstract level for regular expressions, and proposed a new strategy for proving completeness which works uniformly on four axiomatisations. Instead of merging recursive equations as performed in the well-known approach proposed by Milner which usually causes multiple increase of the number of recursive equations, in the new approach we construct canonical solutions, for which one only deals with recursive equations not exceeding the original number. We hope that this will set up a foundation for more feasible implementation of automated proof tools, which after inputting two expressions will automatically output a shorter formal proof of their equality, if any proof exists.

References

1 L. Aceto, R. van Glabbeek, W. Fokkink, and A. IngÍfsdóttir. Axiomatizing prefix itteration with silent steps. *Information and computation*, 127(1):26–40, 1996. doi:10.1006/inco.1996.0047.

2 Y. Deng. A simple completeness proof for the axiomatisations of weak behavioural equivalences. *Bulletin of the European Association for Theoretical Computer Science*, 172:359–397, 2007.

3 M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery*, 32(1):137–161, 1985. doi:10.1145/2455.2460.

4 M. Lohrey, P.R. D'Argenio, and H. Hermanns. Axiomatising divergence. *Information and computation*, 203:115–144, 2005. doi:10.1016/j.ic.2005.05.007.

5 R. Milner. A complete inference system for a class of regular behaviours. *Journal of computer and system sciences*, 28:439–466, 1984. doi:10.1016/0022-0000(84)90023-0.

6 R. Milner. *Communication and Concurrency.* Prentice–Hall, 1989.

7 R. Milner. A complete axiomatisation system for observational congruence of finite-state behaviours. *information and computation*, 81:227–247, 1989. doi:10.1016/0890-5401(89)90070-9.

8 R. J. van Glabbeek. A complete axiomatisation for branching bisimulation congruence of finite-state behaviours. *In A.M Borzyszkowski & S. Sokolowski, editors: Proceedings 18th International Symposium on Mathematical Foundations of Computer Science, MFCS'93, Gdansk, Opland, August/September 1993, LNCS 711, Springer*, pages 473–484, 1993. doi:10.1007/3-540-57182-5_39.

9 R. J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the Association for Computing Machinery*, 43(3):555–600, 1996. doi:10.1145/233551.233556.

10 D. Walker. Bisimulation and divergence. *Information and computation*, 85:220–241, 1990. doi:10.1016/0890-5401(90)90048-M.

A Proof of Theorem 29

In the proof we need to use the lemma of substitution, which we state as follows without a proof. The proof is a routine syntax analysis.

▶ **Lemma 32** (Substitution). *Let $E, F, E_1, \dots, E_n \in \mathcal{E}$, $X, X_1, \dots, X_n$ be variables which are pairwise different. Then the following two equalities hold:*

$$E\{F/X\}\{E_1/X_1, \dots, E_n/X_n\} \equiv E\{F\{E_1/X_1, \dots, E_n/X_n\}/X, E_1/X_1, \dots, E_n/X_n\},$$
$$E\{E_1/X_1, \dots, E_n/X_n\}\{F/X\} \equiv E\{E_1\{F/X\}/X_1, \dots, E_n\{F/X\}/X_n, F/X\}.$$

Proof of Theorem 29. Let us name the recursive specification S. First we prove the following for each E_i by induction on the size of S (i.e. the number of equations in S):

1. for $a \in \mathcal{A}_\tau, E \in \mathcal{E}$, $E_i \xrightarrow{a} E$ if and only if $F_i\{E_1, \dots, E_n/X_1, \dots, X_n\} \xrightarrow{a} E$ and for $W \in \mathcal{V}$, $E_i \rhd W$ if and only if $F_i\{E_1, \dots, E_n/X_1, \dots, X_n\} \rhd W$;
2. $fv(E_i) \subseteq \bigcup\{fv(F_j) \mid 1 \leq j \leq n\} - \{X_1, \dots, X_n\}$.

Once this is established, E_i provably solves $\{X_i = F_i \mid i = 1, \dots, n\}$ in variable X_i follows from 1. and Lemma 14.

For the base case S has just one equation, let E_1 be $\mu X_1.F_1$. Then 1. holds by the fact that for $a \in \mathcal{A}_\tau$, $E \in \mathcal{E}$, $\mu X_1.F_1 \xrightarrow{a} E$ if and only if $F_1\{\mu X_1.F_1/X_1\} \xrightarrow{a} E$, which follows from the operational semantics in Definition 1., 2. holds because $fv(\mu X_1.F_1) = fv(F_1) - \{X_1\}$.

For the induction step, let

$$S' = \{X_i = F_i\{\mu X_n.F_n/X_n\} \mid i = 1, \dots, n-1\}.$$

Then S' is a recursive specification of size $n-1$. By the induction hypothesis there exist $n-1$ expressions $E_1, \dots, E_{n-1}$ such that the following hold for each $1 \le i \le n-1$:

3. for $a \in \mathcal{A}_\tau, E \in \mathcal{E}$, $E_i \xrightarrow{a} E$ iff $F_i\{\mu X_n.F_n/X_n\}\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\} \xrightarrow{a} E$;
4. $fv(E_i) \subseteq \bigcup\{fv(F_j\{\mu X_n.F_n/X_n\}) \mid 1 \le j \le n-1\} - \{X_1, \dots, X_{n-1}\}$.

Now let $E_n \equiv \mu X_n.F_n\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\}$, then by the operational semantics in Definition 1 $E_n \xrightarrow{a} E$ if and only if $F_n\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\}\{E_n/X_n\} \xrightarrow{a} E$. By the lemma of substitution, $F_n\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\}\{E_n/X_n\} \equiv F_n\{E_1/X_1, \dots, E_n/X_n\}$ (note that X_n is not free in $F_j\{\mu X_n.F_n/X_n\}$ for $1 \le j \le n-1$, with condition 4) above X_n is not a free variable in E_i for $i = 1, \dots, n-1$, thus $E_i\{E_n/X_n\} \equiv E_i$). Again by the lemma of substitution, we also have the following equalities for $1 \le i \le n-1$:

$$\begin{aligned}
&F_i\{\mu X_n.F_n/X_n\}\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\} \\
&\equiv F_i\{\mu X_n.F_n\{E_1/X_1, \dots, E_{n-1}/X_{n-1}\}/X_n, E_1/X_1, \dots, E_{n-1}/X_{n-1}\} \\
&\equiv F_i\{E_1/X_1, \dots, E_n/X_n\}
\end{aligned}$$

together with 3), with $E_1, \dots, E_n$ we arrive at the claim 1. for S. Direct calculation gives $fv(F_j\{\mu X_n.F_n/X_n\}) \subseteq (fv(F_j) \cup fv(F_n)) - \{X_n\}$, thus from 4) the following hold for $1 \le i \le n-1$:

$$\begin{aligned}
&fv(E_i) \subseteq \bigcup\{(fv(F_j) \cup fv(F_n)) - \{X_n\} \mid 1 \le j \le n-1\} - \{X_1, \dots, X_{n-1}\} \subseteq \\
&\bigcup\{fv(F_j) \mid 1 \le j \le n\} - \{X_1, \dots, X_n\}.
\end{aligned}$$

Hence we arrive at the claim 2. for S. ◀