Decidable Subsets of CCS

based on the paper with the same title by Christensen, Hirshfeld and Moller from 1994

Sven Dziadek

Abstract

Process algebra is a very interesting framework for describing and analyzing concurrent systems. Therefore CCS is used very often and considered pretty important. But it is undecidable if two CCS agents are bisimilar.

Luckily Christensen, Hirshfeld and Moller showed 1994 in their paper "Decidable Subsets of CCS" that bisimulation for CCS without communication or without restriction and relabelling is decidable. As this subset is powerful enough for many problems, this is an interesting discovery. In addition they derived a complete axiomatisation for these two sublanguages. I want to present this work.

1 Introduction

Over the last 30 years, the study of process algebras has received a great deal of attention. As CCS is one of the first and simplest calculi to analyze processes, it is the basis for most scientific research in this area. Furthermore it is powerful enough to compute any computable function. So it is often used to verify the correctness of systems. This verification uses some kind of equivalence between two processes to prove that a correct process behaves like the one we want to verify. But because CCS is so powerful the equivalence checking is undecidable. This makes verification very hard in general. So the question is in which subsets of CCS the equivalence checking gets decidable. What always works is restricting to finite state systems. But many realistic applications involve infinite state systems.

Milner and Taubner [2, 3] showed that you can reduce the halting problem to equivalence checking if you model Turing machines in full CCS. For their model they had to use communication, restriction and relabelling.

Indeed not only the proof depends on these operations but also sublanguages without communication and sublanguages without restriction and relabelling are decidable. Both sublanguages can express a rich class of infinite state systems. To show the decidability Christensen et al. used a so called tableau decision method invented by Hüttel and Stirling [4].

In the end there follows a complete axiomatisation of these new subsets.

1.1 CCS

To make sure we all work with the same version of CCS I briefly define everything we need.

Let Λ be a set of atomic labels that not contains τ. We define $Act = \Lambda \cup \{\tau\}$. We also need a complementation function $\bar{\cdot} : Act \to Act$ with the properties $\overline{\overline{a}} = a$ and $\overline{\tau} = \tau$. As process variables we use $Var = \{X, Y, Z, ...\}$. CCS expressions are recursively defined as follows:

$$E, F ::= 0 \mid X \mid aE \mid E+F$$

$$\mid E|F \mid E\backslash L \mid E\,[f]$$

For the definition holds that $X \in Var$, $a \in Act$, $L \subseteq \Lambda$ and $f : Act \to Act$ is a relabelling function that satisfies $\overline{f(a)} = f(\overline{a})$. Here is a short description:

- 0 is the nil process which does nothing.
- X is a Variable.
- aE performs first action a and proceeds then with process E.
- $E + F$ means either process E or F will be executed. This is the choice operator.
- $E \mid F$ means process E and F are executed in parallel. They are independent except for complementary actions. This is the composition operator.
- $E\backslash L$ means process E is executed but cannot perform any actions a or $\overline{a}$ with $a \in L$.
- $E\,[f]$ is the process E but all actions get relabeled by function f.

We define a CCS process as a CCS term coupled with a finite family of recursive process equations

$$\Delta = \{X_i \triangleq E_i : 1 \leq i \leq n\}$$

where the X_is are pairwise different and the E_is are CCS expressions that only use the variables $Var(\Delta) = \{X_1, ..., X_n\}$. X_1 is the leading variable.

Another assumption that must be made is that every variable occurrence in the E_is is guarded. That means every variable appears in the scope of an action prefix aE. This is a standard restriction which does not affect the power of the calculus.

Every CCS process determines a labelled transition system. The transition relation is the least relation which preserves the following rules.

$$aE \xrightarrow{a} E \qquad \frac{E \xrightarrow{a} E'}{E \mid F \xrightarrow{a} E' \mid F}$$

$$\frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'} \qquad \frac{E \xrightarrow{a} E',\ F \xrightarrow{\overline{a}} F'}{E \mid F \xrightarrow{\tau} E' \mid F'}\ (a \neq \tau)$$

$$\frac{F \xrightarrow{a} F'}{E \mid F \xrightarrow{a} E \mid F'} \qquad \frac{E \xrightarrow{a} E'}{E\backslash L \xrightarrow{a} E'\backslash L}\ (a, \overline{a} \notin L)$$

$$\frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \qquad \frac{E \xrightarrow{a} E'}{X \xrightarrow{a} E'}\ (X \triangleq E \in \Delta)$$

$$\frac{E \xrightarrow{a} E'}{E\,[f] \xrightarrow{f(a)} E'\,[f]}$$

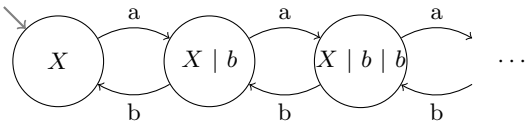
Note that all transition graphs are finite-branching. This is why we restricted to guarded expressions.

To simplify terms we use a congruence relation over expressions. This relation makes e.g. the choice operator commutative and associative. Moreover we do not want to distinguish between $X \mid 0$ or X. We therefore infer all expressions modulo $\equiv$:

Definition 1.1. *Let $\equiv$ be the smallest congruence relation over process expressions such that the laws of associativity, commutativity and 0-absorption hold for choice and composition.*

We will from now on always talk about processes modulo $\equiv$. This means in particular we ignore 0 processes in parallel or in sums. We can do this because bisimilarity (which we will use) satisfies the basic laws of $\equiv$.

Example 1. Here is a short example to demonstrate this congruence relation. Let $\Delta = \{X \triangleq a(X \mid b)\}$. Then X produces this infinite-state transition graph (modulo $\equiv$):



The equivalence relation between CCS expressions which we will use is bisimilarity [5, 2]:

Definition 1.2. *A bisimulation $\mathcal{R}$ is a binary relation over CCS expressions if whenever $E\mathcal{R}F$ then for each $a \in \Lambda$,*

- *if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' with $E'\mathcal{R}F'$*
- *if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' with $E'\mathcal{R}F'$.*

Processes E and F are bisimilar if they are related by some bisimulation. We then write $E \sim F$.

We also want to introduce $Var(\Delta)^{\otimes}$ which is the set of finite multisets over $Var(\Delta) = \{X_1, ..., X_n\}$ and Greek letters $\alpha, \beta, ...$ are the elements of $Var(\Delta)^{\otimes}$. Each such α denotes a CCS process by forming the product of elements of α. Multiplying terms means combining them in parallel using the composition operator. Note that the empty product is 0.

1.2 Standard form

Definition 1.3. *A family $\Delta = \{X_i \triangleq E_i : 1 \leq i \leq n\}$ of CCS equations is in standard form iff every expression E_i is of the form*

$$a_1\alpha_1 + \cdots + a_m\alpha_m$$

with $\alpha_j \in Var(\Delta)^{\otimes}$ for each j.

Again the empty sum is 0.

We want to show that every CCS equation which does not involve restriction/ relabelling or does not involve communication can be converted into standard form. We therefore introduce tree lemmata that shall help to prove this. We first transform processes without communication into processes without communication and restriction. Then we eliminate the relabelling operator in processes without communication and restriction. The third lemma will show how to convert processes without restriction and relabelling into processes in standard form.

For our proofs we need the recursion axioms given by Milner [2]:

Fact 1. *Let $\Delta = \{X_i \triangleq E_i : 1 \leq i \leq n\}$ be an CCS process. Then it holds:*

- *$X_i \sim E_i$ for $1 \leq i \leq n$.*
- *If $P_i \sim E_i\left\{{}^{P_j}/_{X_j}\right\}$ for each $1 \leq i \leq n$ then $P_i \sim X_i$ for each $1 \leq i \leq n$.*

$E_i\left\{{}^{P_j}/_{X_j}\right\}$ describes the process E_i where all occurrences of X_j are substituted by P_j for all j with $1 \leq j \leq n$.

Lemma 1.1. *Let $\Delta = \{X_i \triangleq E_i : 1 \leq i \leq n\}$ be an CCS process which does not involve communication (which means that composition is given by communication-free merge). Then we can effectively construct another CCS process Δ' which also does not involve the restriction operator and for which $\Delta \sim \Delta'$, which means their leading variables are bisimilar.*

Proof. Let $\mathcal{L} = \{L_1, L_2, \ldots, L_k\}$ be the smallest set of restriction sets which is closed under union and contains the empty set $L_1 = \emptyset$, contains the restriction sets appearing in Δ and contains $f^{-1}(L)$ for all L that are already contained and for all f in Δ.

We use some sound axioms as rewrite rules to push the restrictions in CCS expressions down to variable level:

$$\begin{aligned}
0\backslash L &\sim 0 \\
(aE)\backslash L &\sim \begin{cases} aE\backslash L, & \text{if } a, \bar{a} \notin L \\ 0, & \text{otherwise} \end{cases} \\
(E + F)\backslash L &\sim E\backslash L + F\backslash L \\
(E \mid F)\backslash L &\sim E\backslash L \mid F\backslash L \\
E\backslash M\backslash L &\sim E\backslash(M \cup L) \\
E\,[f]\,\backslash L &\sim E\backslash f^{-1}(L)\,[f]
\end{aligned}$$

These rules are all sound. Notice that distributivity over composition is only sound because we only consider processes without communication.

Now let

$$\Delta' = \{Y_{ij} \triangleq F_{ij} : 1 \le i \le n, 1 \le j \le k\}$$

where for each s and t, $F_{st}\left\{{}^{X_i\backslash L_j}/_{Y_{ij}}\right\}$ is the term $E_s\backslash L_t$ rewritten with the above rules.

By part 1 of Fact 1 we know that $X_s \sim E_s$. As bisimilarity is a congruence relation this implies $X_s\backslash L_t \sim E_s\backslash L_t$. Now it follows by the definition of F_{st} and the soundness of the rewrite rules that $X_s\backslash L_t \sim F_{st}\left\{{}^{X_i\backslash L_j}/_{Y_{ij}}\right\}$ for any s and t. So it holds $X_s\backslash L_t \sim Y_{st}$ for any s and t by part 2 of Fact 1. In particular, $X_1 \sim X_1\backslash\emptyset \sim Y_{11}$ which is the same as $\Delta \sim \Delta'$. □

Lemma 1.2. *Let $\Delta = \{X_i \triangleq E_i : 1 \le i \le n\}$ be an CCS process which does not involve communication and does not involve the restriction operator. Then we can effectively construct an equivalent CCS process Δ' which also does not involve the relabelling operator.*

Proof. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$ be the smallest set of relabelling function which contains the identity function $f_1 = \mathrm{id}$, contains all functions appearing in Δ and is closed under composition.

Again we use the some sound axioms as rewrite rules. This time we push the relabellings in CCS expressions down to variable level:

$$\begin{array}{rcl} 0\,[f] & \sim & 0 \\ (aE)\,[f] & \sim & f(a)E\,[f] \\ (E+F)\,[f] & \sim & E\,[f] + F\,[f] \\ (E \mid F)\,[f] & \sim & E\,[f] \mid F\,[f] \\ E\,[g]\,[f] & \sim & E\,[g \circ f] \end{array}$$

Again soundness can easily be seen except for the rule for the composition operator. This rule would be wrong if the relabelling function would create any communication actions but as we excluded communication in our process this rule is sound. Note that we do not need a rule for the restriction operator because this one is also excluded.

Now let

$$\Delta' = \{Y_{ij} \triangleq F_{ij} : 1 \le i \le n, 1 \le j \le k\}$$

where for each s and t, $F_{st}\left\{{}^{X_i[f_j]}/_{Y_{ij}}\right\}$ is the term $E_s\,[f_t]$ which was previously rewritten with the above rules.

By part 1 of Fact 1 and because bisimilarity is a congruence relation it holds $X_s\,[f_t] \sim E_s\,[f_t]$. That implies by the definition of F_{st} and the soundness of the rewrite rules that $X_s\,[f_t] \sim F_{st}\left\{{}^{X_i[f_j]}/_{Y_{ij}}\right\}$ for any s and t. So it follows $X_s\,[f_t] \sim Y_{st}$ for any s and t by part 2 of Fact 1. In particular, $X_1 \sim X_1\,[\mathrm{id}] \sim Y_{11}$ which is the same as $\Delta \sim \Delta'$. □

Example 2. Let f_2 be the function that substitutes a by b and b by a and let $\Delta = \{X \triangleq aX[f_2]\}$.

Now $\mathcal{F} = \{\mathrm{id}, f_2\}$ and

$$\Delta' = \{Y_{11} \triangleq aY_{12}, Y_{12} \triangleq bY_{11}\}$$

because $E_1[f_1] = aX[f_2]$ and $E_1[f_2] = (aX[f_2])[f_2] \sim b(X[f_2][f_2]) \sim bX[f_1]$. Δ' is the relabelling-free equivalent of Δ.

Lemma 1.3. *For any finite family of guarded CCS equations $\Delta = \{X_i \triangleq E_i : 1 \le i \le n\}$ which does not involve restriction and relabelling we can effectively construct another equivalent CCS process Δ' in standard form.*

Proof. Let $\mathcal{E}$ be the set of all subterms of the E_is together with the nil process 0 and the X_is themselves. Now let Y_E be a variable for each $E \in \mathcal{E}$. We will use a function unroll over $\mathcal{E}$ defined as follows:

$$\begin{array}{rcl} \mathrm{unroll}(0) & = & 0 \\ \mathrm{unroll}(X_i) & = & \mathrm{unroll}(E_i) \\ \mathrm{unroll}(aE) & = & aY_E \\ \mathrm{unroll}(E+F) & = & \mathrm{unroll(E)} + \mathrm{unroll(F)} \\ \mathrm{unroll}(E \mid F) & = & \end{array}$$

$$\begin{array}{rcl} \sum a_i(\alpha_i \mid Y_F) + & & \sum b_j(Y_E \mid \beta_j) \\ & + & \sum_{a_i = \bar{b}_j} \tau(\alpha_i \mid \beta_j) \end{array}$$

$$\text{where} \sum a_i\alpha_i = \text{unroll}(E)$$
$$\text{and} \sum b_j\beta_j = \text{unroll}(F)$$

The rule for the composition is exactly the expansion law for composition [2]. It respects the possibility for a potential communication action. (See Example 3.) As we only consider guarded expressions, this function is well-defined.

By induction we can show that unroll(E) is in standard form. We therefore use an ordering on CCS expression which is like the structural ordering with the exceptions that $X_i > E_i$ and aE is a base case for all CCS expressions E. Notice that the exception $X_i > E_i$ does not induce any circles because we restrict on guarded expressions and therefore we will never find any unguarded variable X in the E_is but we can find an expression aE where X_i is a subexpression of E. But as our induction will use aE as a base case this is no problem. On this ordering the induction is easy to verify.

Let

$$\Delta' = \{Y_E \triangleq \text{unroll}(E) : E \in \mathcal{E}\}$$

By the same induction as above we can also show that for any $E \in \mathcal{E}, Y_E \sim E\left\{{}^{Y_{X_i}}/_{X_i}\right\}$. Just use repeatedly the first part of Fact 1.

We further now by part 1 of Fact 1 that $Y_{X_i} \sim \text{unroll}(X_i)$. By definition of the function unroll, $\text{unroll}(X_i) = \text{unroll}(E_i)$. Again the first part of Fact 1 tells us that $\text{unroll}(E_i) \sim Y_{E_i}$. We know from above that $Y_{E_i} \sim E_i\left\{{}^{Y_{X_i}}/_{X_i}\right\}$. By transitivity we get $Y_{X_i} \sim E_i\left\{{}^{Y_{X_i}}/_{X_i}\right\}$. This gives us by the second part of Fact 1, $Y_{X_i} \sim X_i$. In particular $Y_{X_1} \sim X_1$ and therefore $\Delta \sim \Delta'$. □

Example 3. This small example shall demonstrate the expansion law for composition. Let $\{X_1 \triangleq aX_3, X_2 \triangleq \bar{a}X_4\}$ be the (interesting) part of a family of process equations. Then

$$\text{unroll}(X_1) = \text{unroll}(aX_3) = aY_{X_3}$$

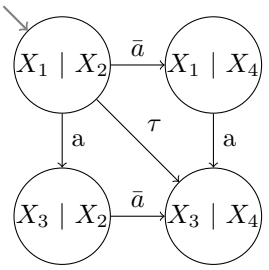
and

$$\text{unroll}(X_2) = \text{unroll}(\bar{a}X_4) = \bar{a}Y_{X_4}$$

Therefore

$$\begin{aligned}\text{unroll}(X_1 \mid X_2) = a(Y_{X_3} \mid Y_{X_2}) \\ + \bar{a}(Y_{X_1} \mid Y_{X_4}) \\ + \tau(Y_{X_3} \mid Y_{X_4})\end{aligned}$$

Compare this result to the first possible transitions of $X_1 \mid X_2$:



Example 4. Here is an example that will convert the CCS process from Example 1 into standard form. Let $\Delta = \{X \triangleq a(X \mid b)\}$. Then $\mathcal{E} = \{0, X, X|b, b\}$. We further know $\text{unroll}(X) = \text{unroll}(a(X \mid b)) = aY_{X|b}$, $\text{unroll}(b) = bY_0$ and $\text{unroll}(X \mid b) = a(Y_{X|b} \mid Y_b) + b(Y_X \mid Y_0)$. It follows:

$$\begin{aligned}\Delta' = \{Y_0 \triangleq 0, Y_x \triangleq aY_{X|b}, Y_b \triangleq bY_0, \\ Y_{X|b} \triangleq a(Y_{X|b} \mid Y_b) + b(Y_x \mid Y_0)\}\end{aligned}$$

Theorem 1.1. *Given any finite family of guarded CCS equations Δ which either does not involve restriction and relabelling or does not involve communication, we can effectively construct another finite family of CCS equations Δ' in standard form in which $\Delta \sim \Delta'$.*

Proof. If Δ does not involve restriction and relabelling we use Lemma 1.3 to convert it into a process in standard form. If it does not involve communication, then we use first Lemma 1.1 and then Lemma 1.2 to convert it into a process without restriction and relabelling and afterwards use Lemma 1.3 to convert it into standard form. □

1.3 Ordering on variables

For the proof of decidability of bisimulation we need the following ordering on $Var(\Delta)^{\otimes}$.

Definition 1.4. *Let $\sqsubset$ be the well-founded ordering on $Var(\Delta)^{\otimes}$ which is defined as follows:*

$$X_1^{k_1} \mid X_2^{k_2} \mid \cdots \mid X_n^{k_n} \sqsubset X_1^{l_1} \mid X_2^{l_2} \mid \cdots \mid X_n^{l_n}$$

iff there exists j such that $k_j < l_j$ and for all $i < j$ holds $k_i = l_i$.
X_i^m means k-times X_i in parallel with the composition operator. Remember the empty product is 0.

$\sqsubset$ is well-founded because it is nothing else but the lexical ordering on tuples of natural numbers, so 0 is the smallest process. Furthermore we are interested in the fact that $\sqsubset$ is total which means for all $\alpha, \beta \in Var(\Delta)^{\otimes}$ with $\alpha \neq \beta$ it holds $\alpha \sqsubset \beta$ or $\beta \sqsubset \alpha$. This is the case because $<$ over positive integers is total and therefore the lexical ordering on tuples is also total. Another important fact is that $\sqsubset$ is a congruence relation which means that $\alpha \sqsubset \beta$ implies $\alpha \mid \gamma \sqsubset \beta \mid \gamma$ for any $\gamma \in Var(\Delta)^{\otimes}$.

2 Decidability of the subset

Assume we have a finite family $\Delta = \{X_i \triangleq E_i : 1 \leq i \leq n\}$ of guarded CCS equations in standard form. We now want to decide for any α and β of $Var(\Delta)^{\otimes}$ if $\alpha \sim \beta$ or not.

2.1 Tableau system

We do this with the tableau decision method invented by Hüttel and Stirling [4]. Other than they, we will use the tree upside-down to avoid confusion when we later see proof trees for the axiomatisation.

A tableau is a tree where all nodes are labelled with equations like $E = F$. The root is the equations we would like to test for bisimilarity. To get easier expressions we always simplify CCS expressions in our tableaux within the rules of the congruence relation $\equiv$. This also avoids confusion when dealing with equal processes that are syntactically different.

Rec	$\dfrac{\mathrm{unf}(\alpha) = \mathrm{unf}(\beta)}{\alpha = \beta}$	
Sum	$\dfrac{\{a_i\alpha_i = b_{f(i)}\beta_{f(i)}\}_{i=1}^{n}\ \{b_j\beta_j = a_{g(j)}\alpha_{g(j)}\}_{j=1}^{m}}{\sum_{i=1}^{n} a_i\alpha_i = \sum_{j=1}^{m} b_j\beta_j}$ with $f : \{1,...,n\} \to \{1,...,m\}$ and $g : \{1,...,m\} \to \{1,...,n\}$	
Prefix	$\dfrac{\alpha = \beta}{a\alpha = a\beta}$	
SubL	$\dfrac{\beta \mid \gamma = \delta}{\alpha \mid \gamma = \delta}$	if the dominated node is labelled $\alpha = \beta$ or $\beta = \alpha$ with $\alpha \sqsupset \beta$
SubR	$\dfrac{\delta = \beta \mid \gamma}{\delta = \alpha \mid \gamma}$	if the dominated node is labelled $\alpha = \beta$ or $\beta = \alpha$ with $\alpha \sqsupset \beta$

Table 1: Rules of the tableau system

Certain rules are applied to the nodes to build up the tableau. These rules are only applied to nodes that are not terminal. Terminal nodes can be either successful or unsuccessful. A successful terminal node is labelled $\alpha = \alpha$ and an unsuccessful terminal node is labelled $a\alpha = 0$ or $0 = a\beta$ or $a\alpha = b\beta$ with $a \neq b$. A tableau is successful if and only if all terminal nodes are successful. Each rule has the form

$$\frac{E_1 = F_1 \qquad \cdots \qquad E_n = F_n}{E = F}$$

The root of the rule represents the goal to be achieved and the children are subgoals that must be shown successful. Table 1 shows the rules.

For the rule Rec we need $\mathrm{unf}(\alpha)$ which means unfolding of α (which is a more general version of the rule $\mathrm{unroll}(E \mid F)$ in the proof of Lemma 1.3). Let $Y_i = \sum_{j=1}^{n_i} a_{ij}\alpha_{ij}$ for $1 \leq i \leq m$. (Notice that Y_i can always be represented in

that way because we only consider CCS equations in standard form.)

$$\mathrm{unf}(\prod_{i=1}^{m} Y_i) = \sum_{i=1}^{m}\sum_{j=1}^{n_i} a_{ij}(\prod_{\substack{k=1\\k\neq i}}^{m} Y_k \mid \alpha_{ij})$$

$$+ \sum_{\substack{i,i'=1\\i\neq i'}}^{m}\sum_{j=1}^{n_i}\sum_{\substack{j'=1\\a_{ij}=\overline{a}_{i'j'}\neq\tau}}^{n_{i'}} \tau(\prod_{\substack{k=1\\k\neq i,i'}}^{m} Y_k \mid \alpha_{ij} \mid \alpha_{i'j'})$$

Furthermore we need some terminology. Tableaux with root $\alpha = \beta$ are indicated by $T(\alpha = \beta)$. Paths are denoted by π and nodes by n or by r if it is the root. If the node is labelled by $E = F$ then we write $n : E = F$.

Basic nodes are nodes of the form $n : \alpha = \beta$. Basic nodes can dominate other basic nodes. A node $n : \alpha \mid \gamma = \delta$ or $n : \delta = \alpha \mid \gamma$ dominates any node $n' : \alpha = \beta$ or $n' : \beta = \alpha$ (with $\alpha \sqsupset \beta$) which appears prior to n in the tableau and to which rule Rec is applied. When a basic node dominates a previous one, we apply one of the Sub rules before applying the Rec rule.

Example 5. $\{X_1 \triangleq a(X_1 \mid X_4), X_2 \triangleq aX_3, X_3 \triangleq a(X_3 \mid X_4) + bX_2, X_4 \triangleq b\}$ is a CCS process in standard form. And this is a successful tableau for $X_1 = X_2$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X_3 \mid X_4 = X_3 \mid X_4}{a(X_3 \mid X_4) = a(X_3 \mid X_4)}\,\text{Prefix} \quad \dfrac{X_2 = X_2}{bX_2 = bX_2}\,\text{Prefix}}{a(X_3 \mid X_4) + bX_2 = a(X_3 \mid X_4) + bX_2}\,\text{Sum}}{X_2 \mid X_4 = X_3}\,\text{Rec}}{X_1 \mid X_4 = X_3}\,\text{SubL}}{a(X_1 \mid X_4) = aX_3}\,\text{Prefix}}{X_1 = X_2}\,\text{Rec}$$

2.1.1 Finiteness

Lemma 2.1. *Every tableau for $\alpha = \beta$ is finite. Additionally there are only finitely many tableaux for $\alpha = \beta$.*

Proof. Let $T(\alpha = \beta)$ be a tableau. Assume for a contradiction that it is infinite. It can only be infinite if it has an infinite path because every node has finite degree. Suppose π is the infinite path starting at the root $r : \alpha = \beta$. π must contain infinitely many basic nodes to which rule Rec is applied because Sum and Prefix only make the expressions smaller and can alone only finitely often be used and the Sub rules can also only finitely often be applied because they are bounded by the relation $\sqsubset$ which is well-founded.

Let $S = \{n_i : \alpha_i = \beta_i\}_{i=1}^{\infty}$ be a sequence of the basic nodes on path π to which rule Rec is applied. So n_1 is the root, n_2 is the second basic node on the path π, and so on. Note that every $\alpha = \beta$ which is nothing else then

$$X_1^{k_1} \mid \cdots \mid X_n^{k_n} = X_1^{l_1} \mid \cdots \mid X_n^{l_n}$$

can be written as a vector

$$\tilde{u} = \begin{pmatrix} k_1 & \cdots & k_n & l_1 & \cdots & l_n \end{pmatrix}^{\top} \in \mathbb{N}^{2n}.$$

So we can rewrite sequence S by a sequence of vectors $\{\tilde{u}_i\}_{i=1}^{\infty}$ with $\tilde{u}_i \in \mathbb{N}^{2n}$ for every i. The first n coordinates of $\tilde{u}_i$ represent α_i and the last n coordinates represent β_i. Now consider the sequence $\{\tilde{u}_i(1)\}_{i=1}^{\infty}$ of the first coordinates of S. This sequence contains either an infinite constant subsequence (if the sequence is upper bounded) or an infinite non-decreasing subsequence, which means $\tilde{u}_i(1) \leq \tilde{u}_j(1)$ for all $i \leq j$ (if the sequence is not bounded). We extract from S the related subsequence with the property that its first coordinate is non-decreasing. Now we continue with the second coordinate and so on. In the end we arrive at an infinite sequence $\{\tilde{w}_i\}_{i=1}^{\infty}$ where all coordinates are non-decreasing.

But then every node in this sequence is dominated by all nodes after it. This comes from the fact that in a non decreasing sequence where α_i occurs before α_j there exists γ for which $\alpha_i = \alpha_j \mid \gamma$. Let us go back to vectors to see this: If $\alpha_i \mathrel{\widehat{=}} \tilde{v}_i$ and $\alpha_j \mathrel{\widehat{=}} \tilde{v}_j$ (with $\tilde{v}_k \in \mathbb{N}^n$) then $\gamma \mathrel{\widehat{=}} \tilde{v}_j - \tilde{v}_i$. As every coordinate was non-decreasing it follows that every coordinate

in γ is non-negative and therefore it is a valid process. Knowing that we can construct any node of this sequence as a product of a previous node and another process, it is easy to see that there is always a Sub rule applicable. But if a Sub rule is applicable that means that it must be applied and the rule Rec cannot be applied to any of these nodes.

It remains to show that there are only finitely many tableaux for $\alpha = \beta$. Assume for a contradiction that there were infinitely many tableaux for $\alpha = \beta$. As for any finite size there can only be finitely many tableaux, the size cannot be bounded. So there must exists an infinite sequence of tableaux where the tableaux are increasing in size. These tableaux must be build up of a partial tableau to which we can infinitely often apply some of the rules to create theses larger and larger tableaux. But then we must be able to apply these rules infinitely often which means we get an infinite path through the tableau. But that cannot be as showed above. □

2.1.2 Completeness

Theorem 2.1. *If $\alpha \sim \beta$ then there exists a successful tableau with root labelled $\alpha = \beta$.*

Proof. Let $\alpha \sim \beta$. We know that we can construct a tableau $T(\alpha = \beta)$. By Lemma 2.1 we know that this tableau will be finite so our construction terminates. Now we have to show that we can apply the rules in a way that every node $N : E = F$ satisfies $E \sim F$. Since if this is the case, every terminal will be successful and therefore the tableau will be successful.

It is clear that the root $r : \alpha = \beta$ indeed satisfies $\alpha \sim \beta$ as this is what we assumed. If now every rule is sound, we can show inductively that the consequent nodes must also satisfies that property. The rules Prefix and Sum can easily verified to be correct, the rule Rec reflects the expansion law for composition [2] which we already saw in the proof for the standard form and the Sub rules follow from the property of bisimilarity to be a congruence relation. □

2.1.3 Soundness

The proof of soundness of the tableau system uses another characterization of bisimulation, namely the sequence of approximations.

Definition 2.1. *The sequence of bisimulation approximations $\{\sim_n\}_{n=0}^{\infty}$ is defined inductively as follows.*

- *$E \sim_0 F$ for all processes E and F;*
- *$E \sim_{n+1} F$ iff for each $a \in \Lambda$,*
  - *if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' with $E' \sim_n F'$;*
  - *if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' with $E' \sim_n F'$.*

Milner proved [2] that for finite-branching transition graphs (our graphs are finite-branching because we disallowed unguarded expressions), bisimulation is the limit of the above approximations:

$$\sim = \bigcap_{n=0}^{\infty} \sim_n$$

Actually he showed that for all $\lambda > k$ follows $\sim_\lambda \subseteq \sim_k$. So the relation $\sim_k$ decreases non-strictly as k increases. As $\sim_k$ is bounded from below by the empty relation $\emptyset$ and $\sim_0$ can be chosen finite, as it must only contain the CCS expressions in our finite-branching transition graph, there must be a value ω for which $\sim_\lambda \; = \; \sim_\omega$ for each $\lambda > \omega$. This limit is $\sim$.

Theorem 2.2. *If there is a successful tableau for $\alpha = \beta$ then $\alpha \sim \beta$.*

Proof. Assume that $\alpha \nsim \beta$ and $T(\alpha = \beta)$ is a tableau for $\alpha = \beta$. We construct a path

$\pi = \{n_i : E_i = F_i\}$ through this tableau starting at the root $n_1 : \alpha = \beta$.

On our way through the nodes of path π we also construct a sequence of integers $\{m_i : E_i \nsim_{m_i} F_i$ and $E_i \sim_j F_i$ for all $j < m_i\}$. So this sequence describes for every node up to which value the bisimulation approximation holds. As we know that $\alpha \nsim \beta$, the value m_1 of our sequence must be finite. We will then see that we can chose the nodes so that this sequence is non-increasing. Together this will show that for each i holds $E_i \nsim F_i$. So we will find a terminal which is not successful and therefore also the tableau is not successful.

Let $n_i : E_i = F_i$ be the current node on the path and m_i the corresponding value from the sequence. Now we construct $n_{i+1} : E_{i+1} = F_{i+1}$ and m_{i+1} like this:

- If we can apply Sum to n_i then we will get several successor. For at least one of them must hold $E_{i+1} \nsim_{m_i} F_{i+1}$ (otherwise we would have $E_i \sim_{m_i} F_i$). We take this one as new node n_{n+1}. That means $m_{i+1} \leq m_i$.
- If Prefix is applied to n_i then the next node is n_{i+1}. $m_{i+1} = m_i - 1$ because this correspond to one recursion step in the Definition 2.1.
- If we apply Rec to n_i then the next node is the one that is produced and $m_{i+1} = m_i$.
- If SubL is applied then $E_i = F_i$ must be of the form $\alpha \mid \gamma = \delta$ and the dominated node is $n_j : \alpha = \beta$ with $\alpha \sqsupset \beta$. As between these two nodes there must be at least one application of the rule Prefix, it holds that $m_i < m_j$. We take $n_{n+1} : \beta \mid \gamma = \delta$. To show that $m_{i+1} \leq m_i$ it suffice to show that $\beta \mid \gamma \nsim_{m_i} \delta$. And this follows from $\alpha \sim_{m_i} \beta$ and $\alpha \mid \gamma \nsim_{m_i} \delta$ because $\beta \mid \gamma \sim_{m_i} \delta$ together with $\alpha \sim_{m_i} \beta$ would otherwise have implied $\alpha \mid \gamma \sim_{m_i} \delta$. SubR works similar.

This shows that for the terminal $\{n_t : E_t = F_t\}$ it must also hold $E_t \nsim_{m_t} F_t$ with $m_t \leq m_1$. So the terminal is unsuccessful and therefore also the tableau. □

2.2 Decidability

Theorem 2.3. *bisimulation equivalence is decidable on CCS processes which either do not involve the restriction and relabelling operators or do not involve communication.*

Proof. As seen in Theorem 1.1, we can convert CCS processes which either do not involve the restriction and relabelling operators or do not involve communication, into standard form. For CCS processes in standard form we can generate tableaux. If we find a successful tableau we simply answer 'yes'. As there are only finite number of them we can stop when they have all been listed and answer 'no'. By soundness and completeness of the tableau system, we will always give the right answer. □

3 Axiomatisation of the subset

At the end we will talk about an equational theory for CCS processes. We will only use CCS processes without restriction and relabelling or without communication. Therefore we can restrict on processes in standard form as Theorem 1.1 showed. Let Δ be such a CCS process in standard form. The theory will be parameterized by Δ.

3.1 Axioms

Will will use statements like $\Gamma \vdash E = F$ where Γ is a set of assumptions of the form $\alpha = \beta$ and E and F are CCS expressions. The semantic interpretation of a sequent $\Gamma \vdash E = F$, which we write as $\Gamma \models E = F$, is as follows: if for all $(\alpha = \beta) \in \Gamma$ holds $\alpha \sim \beta$, then $E \sim F$. We

Equivalence

R1 $\Gamma \vdash E = E$

R2 $\dfrac{\Gamma \vdash F = E}{\Gamma \vdash E = F}$

R3 $\dfrac{\Gamma \vdash E = F \qquad \Gamma \vdash F = G}{\Gamma \vdash E = G}$

Congruence

R4 $\dfrac{\Gamma \vdash E = F}{\Gamma \vdash aE = aF}$

R5 $\dfrac{\Gamma \vdash E_1 = F_1 \qquad \Gamma \vdash E_2 = F_2}{\Gamma \vdash E_1 + E_2 = F_1 + F_2}$

R6 $\dfrac{\Gamma \vdash E_1 = F_1 \qquad \Gamma \vdash E_2 = F_2}{\Gamma \vdash E_1 \mid E_2 = F_1 \mid F_2}$

Axioms

R7 $\Gamma \vdash E + (F + G) = (E + F) + G$

R8 $\Gamma \vdash E + F = F + E$

R9 $\Gamma \vdash E + E = E$

R10 $\Gamma \vdash E + 0 = E$

R11 $\Gamma \vdash E \mid (F \mid G) = (E \mid F) \mid G$

R12 $\Gamma \vdash E \mid F = F \mid E$

R13 $\Gamma \vdash E \mid 0 = E$

Assumption Introduction

R14 $\Gamma, \alpha = \beta \vdash \alpha = \beta$

Assumption Elimination

R15 $\dfrac{\Gamma, \alpha = \beta \vdash \mathrm{unf}(\alpha) = \mathrm{unf}(\beta)}{\Gamma \vdash \alpha = \beta}$

Table 2: Axiomatization

write $\vdash E = F$ and $\models E = F$ for $\emptyset \vdash E = F$ and $\emptyset \models E = F$.

Table 2 shows the axioms and inference rules.

Definition 3.1. *A proof of* $\Gamma \vdash E = F$ *is a proof tree with root labelled* $\Gamma \vdash E = F$. *All leaves in this tree must be instances of the axioms R1 and R7-R14. The inference rules R2-R6 and R15 specify the relation between a node and its children.*

3.2 Soundness

Theorem 3.1. *If* $\Gamma \vdash E = F$ *then* $\Gamma \models E = F$. *In particular, if* $\vdash E = F$ *then* $E \sim F$.

Proof. Let $\alpha \sim \beta$ for all $(\alpha = \beta) \in \Gamma$, but $E \nsim F$. Assume for a contradiction that T is a proof tree for $\Gamma \vdash E = F$.

We construct a maximal path $\pi = \{\Gamma_i \vdash E_i = F_i\}$ from the root to one of the leaves. We construct it so that $E_i \sim F_i$ holds for all i on the path. This is possible because if it were not, we would get a tree in which all children consist of two bisimilar expressions E_i and F_i. But that implies (if we follow the way back to the root) that the root also consists of two bisimilar processes which is false. In addition we construct a sequence $\{m_i : E_i \nsim_{m_i} F_i$ and $E_i \sim_{m_i - 1} F_i\}$ on the way. This sequence will be non-increasing and strictly decreasing through applications of R4.

The leaf in our path π, say $\Gamma_l \vdash E_l = F_l$ must be an instance of the axiom R14 (of the form $\Gamma', \alpha = \beta \vdash \alpha = \beta$) because all other axioms would imply $E_l \sim F_l$.

In the beginning it must hold $(\alpha = \beta) \notin \Gamma$ because we assumed every equation in Γ is true but the equation in the leaf of path π cannot be true. To insert the assumption $\alpha = \beta$ to Γ we needed an application of rule R15. We say this application is at level j. Between the leaf and this node there must be also one instance of R4 because R15 produces guarded expressions but the axiom is for unguarded expressions. We

say this application is at level k. So we get the following path:

$$\begin{array}{lcl} \text{R14} & \Gamma', \alpha = \beta \vdash \alpha = \beta & \text{(level l)} \\ & \vdots & \\ \text{R4} & \dfrac{\Gamma', \alpha = \beta \vdash \alpha' = \beta'}{\Gamma', \alpha = \beta \vdash a\alpha' = a\beta'} & \text{(level k)} \\ & \vdots & \\ \text{R15} & \dfrac{\Gamma'', \alpha = \beta \vdash \mathrm{unf}(\alpha) = \mathrm{unf}(\beta)}{\Gamma'' \vdash \alpha = \beta} & \text{(level j)} \\ & \vdots & \\ & \Gamma \vdash E = F & \text{(level 1)} \end{array}$$

Now we look back to our sequence. Because of the application of rule R4 we know that $m_l < m_j$. The value m_l implies that $\alpha \nsim_{m_l} \beta$ because $\alpha = \beta$ is the equation at level l. But the sequence also tells us that $\alpha \sim_{m_j - 1} \beta$ because of the node at level j. As $m_l \leq m_j - 1$ follows $\alpha \sim_{m_l} \beta$ which is a contradiction. □

3.3 Completeness

Definition 3.2. *For any node n of a tableau, Recnodes(n) denotes the set of labels of the nodes above n to which the rule Rec is applied. In particular, Recnodes(r) = ∅ where r is the root of the tableau.*

Theorem 3.2. *If $\alpha \sim \beta$ then $\vdash \alpha = \beta$.*

Proof. Let $\alpha \sim \beta$. Then there exists a successful finite tableau $T(\alpha = \beta)$. We show that for any node $n : E = F$ of the tableau holds $\mathrm{Recnodes}(n) \vdash E = F$.

We show this by induction on the depth of the subtableau with the current node as root. Since we built the tableau modulo the congruence relation $\equiv$ we will assume that the axioms R7, R8 and R10-R13 are used whenever they are needed.

If n is a terminal then this must be a successful terminal of the form $\alpha = \alpha$. Therefore $\mathrm{Recnodes}(n) \vdash E = F$ follows from Rule R1.

Now let $n : E = F$ be not a terminal node. We distinct some cases:

- If $E = F$ is of the form $a\gamma = a\delta$ then the rule Prefix is applied. By the induction hypothesis on the child n' we know that $\mathrm{Recnodes}(n') \vdash \gamma = \delta$ holds. As $\mathrm{Recnodes}(n) = \mathrm{Recnodes}(n')$ it follows by rule R4 that $\mathrm{Recnodes}(n) \vdash a\gamma = a\delta$

- If $E = F$ is of the form $\sum_{i=1}^{n} a_i\alpha_i = \sum_{j=1}^{m} b_j\beta_j$ than rule Sum is applied. By induction we know for all children n_k : $a_{i_k}\alpha_{i_k} = b_{j_k}\beta_{j_k}$ that $\mathrm{Recnodes}(n_k) \vdash a_{i_k}\alpha_{i_k} = b_{j_k}\beta_{j_k}$. As $\mathrm{Recnodes}(n) = \mathrm{Recnodes}(n')$ if follows from rule R5, R7, R8 and R9 that $\mathrm{Recnodes}(n) \vdash E = F$.

- If $E = F$ is of the form $\gamma = \delta$ and its son n' is $\mathrm{unf}(\gamma) = \mathrm{unf}(\delta)$ than the rule Rec was applied. By the induction hypothesis holds $\mathrm{Recnodes}(n') \vdash \mathrm{unf}(\gamma) = \mathrm{unf}(\delta)$. As $\mathrm{Recnodes}(n) = \mathrm{Recnodes}(n') \cup \{(\gamma = \delta)\}$ it follows from rule R15 that $\mathrm{Recnodes}(n) \vdash E = F$.

- If $E = F$ is of the form $\mu \mid \gamma = \delta$ and the dominated node is $n' : \mu = \eta$ with $\eta \sqsubset \mu$. So SubL was applied. The son n'' is labelled $\eta \mid \gamma = \delta$. By induction we know $\mathrm{Recnodes}(n'') \vdash \eta \mid \gamma = \delta$. As $\mathrm{Recnodes}(n) = \mathrm{Recnodes}(n'')$ and $(\mu = \eta) \in \mathrm{Recnodes}(n)$, if follows by rule R14, R3, R6 and R1 that $\mathrm{Recnodes}(n) \vdash \mu \mid \gamma = \delta$. SubR works similar.

Since $\mathrm{Recnodes}(r) = \emptyset$ for the root r, $\mathrm{Recnodes}(r) \vdash E = F$, what we just showed, implies $\vdash \alpha = \beta$ what we wanted to show. □

Example 6. Let $\{X_1 \triangleq a(X_1 \mid X_4), X_2 \triangleq aX_3, X_3 \triangleq a(X_3 \mid X_4) + bX_2, X_4 \triangleq b\}$ be a CCS process. (It is the same process as in Example 5.) Table 3 contains a proof for $X_1 = X_2$. To make it easier to read we use these abbreviations: $\Gamma_1 = \{X_1 = X_2\}$ and $\Gamma_2 = \Gamma_1 \cup \{X_2 \mid X_4 = X_3\}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma_1 \vdash X_1 = X_2 \text{ (R14)} \qquad \Gamma_1 \vdash X_4 = X_4 \text{ (R1)}}{\Gamma_1 \vdash X_1 \mid X_4 = X_2 \mid X_4}\,\text{R6} \qquad \dfrac{\dfrac{\dfrac{\Gamma_2 \vdash X_3 \mid X_4 = X_3 \mid X_4 \text{ (R1)}}{\Gamma_2 \vdash a(X_3 \mid X_4) = a(X_3 \mid X_4)}\,\text{R4} \qquad \dfrac{\Gamma_2 \vdash X_2 = X_2 \text{ (R1)}}{\Gamma_2 \vdash bX_2 = bX_2}\,\text{R4}}{\Gamma_2 \vdash a(X_3 \mid X_4) + bX_2 = a(X_3 \mid X_4) + bX_2}\,\text{R5}}{\Gamma_1 \vdash X_2 \mid X_4 = X_3}\,\text{R15}}{\Gamma_1 \vdash X_1 \mid X_4 = X_3}\,\text{R3}}{\Gamma_1 \vdash a(X_1 \mid X_4) = aX_3}\,\text{R4}}{\vdash X_1 = X_2}\,\text{R15}$$

Table 3: Proof tree for $X_1 = X_2$

4 Conclusion

CCS allows the description of any computable function and is therefore universal. But as any considerable equivalence problem is undecidable, full CCS is not the right choice for many applications. For some of these applications it is enough to use only finite state systems where the equivalence problem is easily proved to be decidable.

But most realistic functions and processes involve infinite state systems. This is where the topic of this paper gets interesting. Both, CCS without restriction and relabelling and CCS without communication describe a rich class of infinite state systems where equivalence checking is decidable.

So Christensen et al. did a good job by showing that these two sublanguages are decidable. But it is worth noting that this work is based on another paper of them from 1993 [6] where they showed that CCS without restriction, relabelling and communication is decidable. And their ideas are similar in spirit to the work by Hüttel and Stirling [4] on context-free processes. But remember that context-free processes and CCS without restriction, relabelling and communication can both express processes that cannot be expressed by the other language.

References

[1] S. Christensen, Y. Hirshfeld, F. Moller. Decidable Subsets of CCS. 1994

[2] R. Milner. Communication and Concurrency. 1989

[3] D. Taubner. Finite representations of CCS and TCSP programs by automata and Petri nets. 1989

[4] H. Hüttel, C. Stirling. Actions speak louder than words: proving bisimilarity for context-free processes. 1991

[5] D. Park. Concurrency and automata on infinite sequences. 1981

[6] S. Christensen, Y. Hirshfeld, F. Moller. Bisimulation equivalence is decidable for basic parallel processes. 1993