Further Formalization of the Process Algebra CCS in HOL4

Chun Tian

Scuola di Scienze, Università di Bologna
chun.tian@studio.unibo.it
Numero di matricola: 0000735539

Abstract. In this project, we have extended previous work on the formalization of the process algebra CCS in HOL4, with full supports of weak equivalence, rooted weak equivalence and related definitions and algebraic laws. Some deep lemmas were also formally proved in this project, including Deng Lemma, Hennessy Lemma and two versions of the "coarsest congruence contained in weak equivalence". For the last theorem, with the help of theorem prover we have greatly weakened the assumptions required by existing proofs and successfully formalized the full version without any assumption.

1 Introduction

(to be added)

2 Background

The current project is a further extension of a previous project [1] on the formalization of the process algebra CCS in HOL Theorem Prover (HOL4). In the previous work, we have successfully covered the (strong) transitions of CCS processes, strong bisimulation equivalence and all strong algebraic laws including the expansion law. But this is not a complete work, as in reality the model checkings were usually done by checking the (rooted) weak equivalence between a specification and an implementation for the same model, and the implemetation is usually not τ-free. Thus the author needs to further extended this work to make it really useful.

On the other side, almost all these work were derived from an old CCS formalization [2] on Hol88 theorem prover (ancestry of HOL4), done by Monica Nesi during 1992-1995. The related proof scripts mentioned in the publications of Prof. Nesi is not available on Internet, but on June 7, 2016, Professor Nesi sent some old proof scripts to the author in private, soon after the author asked for these scripts in HOL mailing list. But these scripts did not include any formalization for weak equivalence, rooted weak equivalence and other things (e.g. HML) mentioned in her paper. At the begining the author thought that the rest scripts must have been lost, but it turned out that this is not true.

On May 15, 2017, almost immediately after the author announced the finish of the previous project to all related people, Prof. Nesi replied the mail with the following contents:

> "Dear Chun Tian,
> Thanks a lot for your message, I am happy you were successful in your work! I will try and read your report as soon as I can, but in the meantime I would like to point out that my files on weak bisimulation, weak equivalence, observation congruence, modal logic, etc., are not lost. In a mail to you (dated Jun 7th, 2016) I just sent you the first bunch of files to start with. You said you were still learning HOL, so I thought it better not to "flood" you with all my files. I don't know what your plans are now, but I would be glad to send you other files on CCS in HOL if you fancy going on with this work.
> Best regards, Monica"

Then it became obvious that, another further project on this topic should be done in scope of the "tirocinio" (training) project [1] under the supervision of Prof. Roberto Gorrieri. And instead of

[1] It is an obligatory part in the author's study plan of Master degree in Computer Science

creating everything from scretch, we got another bunch of old scripts to start with. This is a great advantage for doing another successful project. After having expressed such willings to Prof. Nesi, finally on June 6, 2017, the author has received all the rest old proof script on the formalization of pure CCS, covering weak bisimulation, weak equivalence, observation congruence and HML. There're totally about 4000 lines of Classic ML code.

But the current project is not simply a porting of the old scripts without any new creation. Instead, now the author has become more experienced in using HOL4 theorem prover, and he has essentially modified many fundamental definitions and have proved some new deep theorems. And with the new definition and HOL4's existing theory library, previous unprovable theorems now become provable. Below is a brief summary of changes and new features comparing with the old work:

1. We have extended the datatype definitions of CCS processes and transition actions, replacing all strings into general type variables. [2] As a result, now it's possible to do reasoning on processes with limited number of actions and constants. In academic, the notation $CCS(a, b)$ is the CCS subcalculus which can use at most h constants and k actions, and some important results hold for only certain CCS subcalculus (e.g. [4]). With the new datatype, now our formalization has the ability to reason on this kind of CCS subclasulus. For almost all theorems and algebraic laws, such a change doesn't affect the proof at all, the only exception is the "coarsest congruence contained in $\approx$" (Theorem 4.5 in [5] or Proposition 3 in Chapter 7 of [6]), in which the assumption is not automatically true if the possible actions were finite. (We present two formal proofs of this theorem with different assumptions in details in this project.)
2. We have completely turned to use HOL4's builtin supports of co-inductive relations (`Hol_coreln`) for defining strong and weak bisimulation equivalences. As a result, many intermediate definitions and theorems towards to the proof of *Property (*) of strong and weak equivalence* [3] are not needed any more, thus removed from our previous project.
3. We have extensively used HOL4's existing relationTheory and the supports of RTC (reflexive transitive closure) for defining the weak transitions of CCS processes. As a result, a large amount of cases and induction theorems were automatically available. It will show that, without these extra theorems (especiall the right induction theorem) it's impossible to prove the transitivity of observation congruence.
4. Some important theorems were not proved in the old work, the notable ones are: 1) the transitivity of observation congruence and 2) the coarsest congruence containing in weak equivalence. In this project, we have finished these proofs with many new related lemmas created and proved. As for the " coarsest congruence containing in weak equivalence" theorem, we have successfully proved a stronger version (without any assumption) for finite-state CCS based on a new proof [7] published by J. R. van Glabbeek in 2005.
5. In addition, we also formally proved the Hennessy Lemma and Deng Lemma (the weak equivalence version), which shows deep relations between weak equivalence and observation congruence. We have used Deng Lemma to prove the hard part of Hennessy Lemma, therefore minimized the related proof scripts.

The old formalization of Hennessy-Milner Logic (for CCS) is not ported into HOL4 in this project, because our focus in current project is mainly at the theorem proving aspects, i.e. the proof of some deep theorems related to weak bisimulation equivalence and observation congruence (rooted weak bisimulation equivalence). Actually, we have put aside one of the initial project goals, i.e. creating a new model checking tool running in HOL theorem prover. Instead, we have focused on pure theorem-proving staff in this project, and deeply researched the current proofs for several imporant theorems and the precise requirements to make these theorems hold.

3 Extended CCS datatypes

The type of CCS processes has been extended with two type variables, α and β, α denotes the type of constants, and β denotes the type of labels. In HOL, such a higher order type is represented

[2] This is not new invention, Prof. Nesi has done a similar change in her formalization of Value-passing CCS in 1999 [3]. But work seems also done in Hol88, and related code is not available on Internet.

[3] HOL theorem names: `PROPERTY_STAR` and `OBS_PROPERTY_STAR`

as "(α, β) CCS". Whenever both type variables were instantiazed as `string`, the resulting type "(string, string) CCS" is equivalent with the old CCS type in previous project. Within the new settings, to represent CCS subcalculus like $CCS(25, 12)$, custom datatypes with limited number of instances must be defined by users[4].

The type of transition labels were extented by type variable β, the resulting new type is "β Label" in HOL. It's important to notice that, for each possible value l of the type β, both "name l" and "coname l" are valid labels, therefore the totally available number of labels are doubled with the cardinality of the set of all possible values of type β. Also noticed that, the invisible action τ is part of the type "β Action", which contains both τ and "β Label" values (wrapped by constructor `label`. Thus if we count the number of all possible *actions* of the type "(α, β) CCS", it should be the doubled cardinality of type α plus one.

Finally, it should be noticed that, in HOL, each valid type must contain at least one instance, thus in the minimal setting, there're still three valid actions: τ, the singleton input action and output action. Whenever a CCS related theorem requires that "there's at least one non-*tau* action", such a requirement can be omitted from the assumptions of the theorem.

4 Weak transitions and the EPS relation

In previous project [1], we have discussed the advantage to use `EPS` and `WEAK_TRANS` (instead of `WEAK_TRACE` used in [5]) for defining weak bisimulation, weak bisimulation equivalence and observation congruence. But we didn't prove any theorem about `EPS` and `WEAK_TRANS` in previous project. In this project, we have slightly changed the definition of `EPS` without the need to define another constant first:

Definition 1. *(EPS) For any two CCS processes $E, E' \in Q$, define relation $EPS \subseteq Q \times Q$ as the reflexive transitive closure (RTC) of single-τ transition between E and E' ($E \xrightarrow{\tau} E'$).*[5]

$\vdash$ `EPS = (λE E'. E --τ-> E')*`

Intuitively speaking, $E \stackrel{\epsilon}{\Rightarrow} E'$ (Math notion: $E \stackrel{\epsilon}{\Longrightarrow} E'$) means there're zero or more tau-transitions from p to q.

Sometimes it's necessary to consider different transition cases when $p \stackrel{\epsilon}{\Rightarrow} q$ holds, or induct on the number of *tau* transitions between p and q. With such a definition, beside the obvious reflexive and transitive properties, a large amount of "cases" and induction theorem already proved in HOL's `relationTheory` are immediately available to us:

Proposition 1. *(The "cases" theorem of the* `EPS` *relation)*

$\vdash x \stackrel{\epsilon}{\Rightarrow} y \iff x = y \vee \exists u.\ x \text{ --}\tau\text{-> } u \wedge u \stackrel{\epsilon}{\Rightarrow} y$ [EPS_cases1]

$\vdash x \stackrel{\epsilon}{\Rightarrow} y \iff x = y \vee \exists u.\ x \stackrel{\epsilon}{\Rightarrow} u \wedge u \text{ --}\tau\text{-> } y$ [EPS_cases2]

$\vdash E \stackrel{\epsilon}{\Rightarrow} E' \iff E \text{ --}\tau\text{-> } E' \vee E = E' \vee \exists E_1.\ E \stackrel{\epsilon}{\Rightarrow} E_1 \wedge E_1 \stackrel{\epsilon}{\Rightarrow} E'$ [EPS_cases]

Proposition 2. *(The induction and strong induction principles of the* `EPS` *relation)*

$\vdash (\forall x.\ P\ x\ x) \wedge (\forall x\ y\ z.\ x \text{ --}\tau\text{-> } y \wedge P\ y\ z \Rightarrow P\ x\ z) \Rightarrow$
$\forall x\ y.\ x \stackrel{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_ind]

$\vdash (\forall x.\ P\ x\ x) \wedge (\forall x\ y\ z.\ x \text{ --}\tau\text{-> } y \wedge y \stackrel{\epsilon}{\Rightarrow} z \wedge P\ y\ z \Rightarrow P\ x\ z) \Rightarrow$
$\forall x\ y.\ x \stackrel{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_strongind]

$\vdash (\forall x.\ P\ x\ x) \wedge (\forall x\ y\ z.\ P\ x\ y \wedge y \text{ --}\tau\text{-> } z \Rightarrow P\ x\ z) \Rightarrow$
$\forall x\ y.\ x \stackrel{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_ind_right]

$\vdash (\forall x.\ P\ x\ x) \wedge (\forall x\ y\ z.\ P\ x\ y \wedge x \stackrel{\epsilon}{\Rightarrow} y \wedge y \text{ --}\tau\text{-> } z \Rightarrow P\ x\ z) \Rightarrow$
$\forall x\ y.\ x \stackrel{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_strongind_right]

$\vdash (\forall E\ E'.\ E \text{ --}\tau\text{-> } E' \Rightarrow P\ E\ E') \wedge (\forall E.\ P\ E\ E) \wedge$
$(\forall E\ E_1\ E'.\ P\ E\ E_1 \wedge P\ E_1\ E' \Rightarrow P\ E\ E') \Rightarrow$
$\forall x\ y.\ x \stackrel{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_INDUCT]

[4] In HOL, there's already a single-instance type `unit`, and a two-valued type `bool`, further custom datatypes can be defined by `Define` command.

[5] In HOL4's `relationTheory`, the relation types is curried: instead of having the same type "α reln" as the math definition, it has the type "α -> α -> bool". And the star(*) notation is for defining RTCs.

Then we define the weak transition between two CCS processes upon the EPS relation:

Definition 2. *For any two CCS processes $E, E' \in Q$, define "weak transition" relation $\Longrightarrow \subseteq Q \times A \times Q$, where A can be τ or a visible action: $E \xrightarrow{a} E'$ if and only if there exists two processes E_1 and E_2 such that $E \overset{\epsilon}{\Longrightarrow} E_1 \xrightarrow{a} E_2 \overset{\epsilon}{\Longrightarrow} E'$:*

$$\vdash E \texttt{ ==}u\texttt{=>> } E' \iff \exists E_1\ E_2.\ E \overset{\epsilon}{\Rightarrow} E_1 \wedge E_1 \texttt{ --}u\texttt{-> } E_2 \wedge E_2 \overset{\epsilon}{\Rightarrow} E' \qquad \texttt{[WEAK\_TRANS]}$$

Using above two definitions and the "cases" and induction theorems, a large amount of properties about EPS and WEAK_TRANS were proved:

Proposition 3. *(Properties of* EPS *and* WEAK_TRANS*)*

1. *Any transition also implies a weak transition:*

$$\vdash E \texttt{ --}u\texttt{-> } E' \Rightarrow E \texttt{ ==}u\texttt{=>> } E' \qquad \texttt{[TRANS\_IMP\_WEAK\_TRANS]}$$

2. *Weak τ-transition implies* EPS *relation:*

$$\vdash E \texttt{ ==}\tau\texttt{=>> } E' \Rightarrow E \overset{\epsilon}{\Rightarrow} E' \qquad \texttt{[WEAK\_TRANS\_TAU]}$$

3. *τ-transition implies* EPS *relation:*

$$\vdash E \texttt{ --}\tau\texttt{-> } E' \Rightarrow E \overset{\epsilon}{\Rightarrow} E' \qquad \texttt{[TRANS\_TAU\_IMP\_EPS]}$$

4. *Weak τ-transition implies an τ transition followed by EPS transition:*

$$\vdash E \texttt{ ==}\tau\texttt{=>> } E' \Rightarrow \exists E_1.\ E \texttt{ --}\tau\texttt{-> } E_1 \wedge E_1 \overset{\epsilon}{\Rightarrow} E' \qquad \texttt{[WEAK\_TRANS\_TAU\_IMP\_TRANS\_TAU]}$$

5. EPS *implies τ-prefixed* EPS*:*

$$\vdash E \overset{\epsilon}{\Rightarrow} E' \Rightarrow \tau..E \overset{\epsilon}{\Rightarrow} E' \qquad \texttt{[TAU\_PREFIX\_EPS]}$$

6. *Weak τ-transition implies τ-prefixed weak: τ-transition:*

$$\vdash E \texttt{ ==}u\texttt{=>> } E' \Rightarrow \tau..E \texttt{ ==}u\texttt{=>> } E' \qquad \texttt{[TAU\_PREFIX\_WEAK\_TRANS]}$$

7. *A weak transition wrapped by EPS transitions is still a weak transition:*

$$\vdash E \overset{\epsilon}{\Rightarrow} E_1 \wedge E_1 \texttt{ ==}u\texttt{=>> } E_2 \wedge E_2 \overset{\epsilon}{\Rightarrow} E' \Rightarrow E \texttt{ ==}u\texttt{=>> } E' \qquad \texttt{[EPS\_AND\_WEAK]}$$

8. *A weak transition after a τ-transition is still a weak transition:*

$$\vdash E \texttt{ --}\tau\texttt{-> } E_1 \wedge E_1 \texttt{ ==}u\texttt{=>> } E' \Rightarrow E \texttt{ ==}u\texttt{=>> } E' \qquad \texttt{[TRANS\_TAU\_AND\_WEAK]}$$

9. *Any transition followed by an EPS transition becomes a weak transition:*

$$\vdash E \texttt{ --}u\texttt{-> } E_1 \wedge E_1 \overset{\epsilon}{\Rightarrow} E' \Rightarrow E \texttt{ ==}u\texttt{=>> } E' \qquad \texttt{[TRANS\_AND\_EPS]}$$

10. *An EPS transition implies either no transition or a weak τ-transition:*

$$\vdash E \overset{\epsilon}{\Rightarrow} E' \Rightarrow E = E' \vee E \texttt{ ==}\tau\texttt{=>> } E' \qquad \texttt{[EPS\_IMP\_WEAK\_TRANS]}$$

11. *Two possible cases for the first step of a weak transition:*

$$\begin{aligned}&\vdash E \texttt{ ==}u\texttt{=>> } E_1 \Rightarrow \\ &\quad (\exists E'.\ E \texttt{ --}\tau\texttt{-> } E' \wedge E' \texttt{ ==}u\texttt{=>> } E_1) \vee \\ &\quad \exists E'.\ E \texttt{ --}u\texttt{-> } E' \wedge E' \overset{\epsilon}{\Rightarrow} E_1 \qquad \texttt{[WEAK\_TRANS\_cases1]}\end{aligned}$$

12. *The weak transition version of SOS inference rule (Sum_1) and (Sum_2):*

$$\vdash E \texttt{ ==}u\texttt{=>> } E_1 \Rightarrow E \texttt{ + } E' \texttt{ ==}u\texttt{=>> } E_1 \qquad \texttt{[WEAK\_SUM1]}$$
$$\vdash E \texttt{ ==}u\texttt{=>> } E_1 \Rightarrow E' \texttt{ + } E \texttt{ ==}u\texttt{=>> } E_1 \qquad \texttt{[WEAK\_SUM2]}$$

5 Weak bisimulation, weak bisimulation equivalence and the algebraic laws

The concepts of weak bisimulation and weak bisimulation equivalence (a.k.a. observation equivalence), together with the algebraic laws for weak bisimulation equivalence, stand at a central position in this project. This is mostly because all the deep theorems (Deng lemma, Hennessy lemma, Coarsest congruence contained in weak equivalence) that we have formally proved in this project, were all talking about the relationship between weak bisimulation equivalence and rooted weak bisimulation equivalence (a.k.a. observation congruence, we'll use this shorted names in the rest of the paper). The other reason is, since the observation congruence is not recursively defined but rely on the definition of weak equivalence, it turns out that, the properties of weak equivalence were heavily used in the proof of properties of observation congruence. Thus from the theorem proving view, the theorems that we're going represent in this section, are the most "useful" theorems depended by the rest work in this project.

On the other side, it's quite easy to derive out almost all the algebraic laws for weak equivalence (and observation congruence), simply because strong equivalence implies weak equivalence (and also observation congruence). This fact also reflects the fact that, although strong equivalence and its algebraic laws were usually useless in real world model checking, they do have contributions for deriving more useful algebraic laws. And from the view of theorem proving it totally make sense: if we try to prove any algebraic law for weak equivalence *directly*, the proof will be quite long and difficult, and the handling of *tau*-transitions will be a common part in all these proofs. But if we use the strong algebraic laws as lemmas, the proofs were actually divided into two logical parts: one for handling the algebraic law itself, the other for handling the τ-transitions.

The definition of weak bisimulation is the same as in [5], except for the use of EPS in case of τ-transitions:

Definition 3. *(Weak bisimulation)*

$$
\begin{array}{l}
\vdash \texttt{WEAK\_BISIM}\ \mathit{Wbsm} \iff \\
\quad \forall E\ E'. \\
\qquad \mathit{Wbsm}\ E\ E' \Rightarrow \\
\qquad (\forall l. \\
\qquad\quad (\forall E_1. \\
\qquad\qquad E\ \texttt{--label}\ l\texttt{->}\ E_1 \Rightarrow \\
\qquad\qquad \exists E_2.\ E'\ \texttt{==label}\ l\texttt{=>>}\ E_2 \wedge \mathit{Wbsm}\ E_1\ E_2) \wedge \\
\qquad\quad \forall E_2. \\
\qquad\qquad E'\ \texttt{--label}\ l\texttt{->}\ E_2 \Rightarrow \\
\qquad\qquad \exists E_1.\ E\ \texttt{==label}\ l\texttt{=>>}\ E_1 \wedge \mathit{Wbsm}\ E_1\ E_2) \wedge \\
\qquad (\forall E_1.\ E\ \texttt{--}\tau\texttt{->}\ E_1 \Rightarrow \exists E_2.\ E' \stackrel{\epsilon}{\Rightarrow} E_2 \wedge \mathit{Wbsm}\ E_1\ E_2) \wedge \\
\qquad \forall E_2.\ E'\ \texttt{--}\tau\texttt{->}\ E_2 \Rightarrow \exists E_1.\ E \stackrel{\epsilon}{\Rightarrow} E_1 \wedge \mathit{Wbsm}\ E_1\ E_2
\end{array}
$$

Weak bisimulation has some common properties:

Proposition 4. *Properties of weak bisimulation*

1. *The identity relation is a weak bisimulation:*

$$\vdash \texttt{WEAK\_BISIM}\ (\lambda x\ y.\ x = y) \qquad \texttt{[IDENTITY\_WEAK\_BISIM]}$$

2. *The converse of a weak bisimulation is still a weak bisimulation:*

$$\vdash \texttt{WEAK\_BISIM}\ \mathit{Wbsm} \Rightarrow \texttt{WEAK\_BISIM}\ (\lambda x\ y.\ \mathit{Wbsm}\ y\ x) \qquad \texttt{[IDENTITY\_WEAK\_BISIM]}$$

3. *The composition of two weak bisimulations is a weak bisimulation:*

$$
\begin{array}{l}
\vdash \texttt{WEAK\_BISIM}\ \mathit{Wbsm}_1 \wedge \texttt{WEAK\_BISIM}\ \mathit{Wbsm}_2 \Rightarrow \\
\quad \texttt{WEAK\_BISIM}\ (\lambda x\ z.\ \exists y.\ \mathit{Wbsm}_1\ x\ y \wedge \mathit{Wbsm}_2\ y\ z) \qquad \texttt{[COMP\_WEAK\_BISIM]}
\end{array}
$$

4. *The union of two weak bisimulations is a weak bisimulation:*

$\vdash$ WEAK_BISIM $Wbsm_1 \wedge$ WEAK_BISIM $Wbsm_2 \Rightarrow$
WEAK_BISIM ($\lambda x\ y.\ Wbsm_1\ x\ y \vee Wbsm_2\ x\ y$) [UNION_WEAK_BISIM]

There're two ways to define weak bisimulation equivalence in HOL4, one is to define it as the union of all weak bisimulations:

Definition 4. *(Alternative definition of weak equivalence) For any two CCS processes E and E', they're* weak bisimulation equivalent *(or weak bisimilar) if and only if there's a weak bisimulation relation between E and E':*

$\vdash E \approx E' \iff \exists Wbsm.\ Wbsm\ E\ E' \wedge$ WEAK_BISIM $Wbsm$ [WEAK_EQUIV]

This is the old method used by Prof. Nesi in Hol88 in which there's no support yet for defining co-inductive relations. The new method we have used in this project, is to use HOL4's new co-inductive relation defining facility `Hol_coreln` to define weak bisimulation equivalence:

```
val (WEAK_EQUIV_rules, WEAK_EQUIV_coind, WEAK_EQUIV_cases) = Hol_coreln ‘
    (!(E :('a, 'b) CCS) (E' :('a, 'b) CCS).
       (!l.
         (!E1. TRANS E  (label l) E1 ==>
               (?E2. WEAK_TRANS E' (label l) E2 /\ WEAK_EQUIV E1 E2)) /\
         (!E2. TRANS E' (label l) E2 ==>
               (?E1. WEAK_TRANS E  (label l) E1 /\ WEAK_EQUIV E1 E2))) /\
       (!E1. TRANS E  tau E1 ==> (?E2. EPS E' E2 /\ WEAK_EQUIV E1 E2)) /\
       (!E2. TRANS E' tau E2 ==> (?E1. EPS E  E1 /\ WEAK_EQUIV E1 E2))
      ==> WEAK_EQUIV E E')‘;
```

The disadvantage of this new method is that, the rules used in above definition actually duplicated the definition of weak bisimulation, while the advantage is that, HOL4 automatically proved an important theorem and returned it as the third return value of above definition. This theorem is also called "the property (*)" (in Milner's book [6]:

Proposition 5. *(The property (*) for weak bisimulation equivalence)*

$\vdash a_0 \approx a_1 \iff$
$(\forall l.$
$\quad (\forall E_1.$
$\quad\quad a_0$ --label l-> $E_1 \Rightarrow$
$\quad\quad \exists E_2.\ a_1$ ==label l=>> $E_2 \wedge E_1 \approx E_2) \wedge$
$\quad \forall E_2.$
$\quad\quad a_1$ --label l-> $E_2 \Rightarrow$
$\quad\quad \exists E_1.\ a_0$ ==label l=>> $E_1 \wedge E_1 \approx E_2) \wedge$
$(\forall E_1.\ a_0$ --τ-> $E_1 \Rightarrow \exists E_2.\ a_1 \stackrel{\epsilon}{\Rightarrow} E_2 \wedge E_1 \approx E_2) \wedge$
$\forall E_2.\ a_1$ --τ-> $E_2 \Rightarrow \exists E_1.\ a_0 \stackrel{\epsilon}{\Rightarrow} E_1 \wedge E_1 \approx E_2$ [OBS_PROPERTY_STAR]

It's known that, above property cannot be used as an alternative definition of weak equivalence, because it doesn't capture all possible weak equivalences. But it turns out that, for the proof of most theorems about weak bisimularities this property is enough to be used as a rewrite rule in their proofs. And, if we had used the old method to define weak equivalence, it's quite difficult to prove above property (*).[6]

Using the alternative definition of weak equivalence, it's quite simple to prove that, the weak equivalence is an equivalence relation:

Proposition 6. *(Weak equivalence is an equivalence relation)*

[6] In our previous project, the property (*) for strong equivalence was proved based on the old method, then in this project we have completely removed these code and now both strong and weak bisimulation equivalences were based on the new method. On the other side, the fact that Prof. Nesi can define co-inductive relation without using `Hol_coreln` has shown that, the core HOL logic doesn't need to be extended to suppport co-inductive relation, and all what `Hol_coreln` does internally is to use the existing HOL theorems to construct the related proofs. This is very different with the situation in other theorem provers (e.g. Coq) in which the core logic has to be extended to support co-induction.

$\vdash$ `equivalence (=~)` [WEAK_EQUIV_equivalence]

or

$\vdash E \approx E$ [WEAK_EQUIV_REFL]
$\vdash E \approx E' \Rightarrow E' \approx E$ [WEAK_EQUIV_SYM]
$\vdash E \approx E' \wedge E' \approx E'' \Rightarrow E \approx E''$ [WEAK_EQUIV_TRANS]

The substitutivity of weak equivalence under various CCS process operators were then proved based on above definition and property (*). However, as we know weak equivalence is not a congruence, in some of these substitutivity theorems we must added extra assumptions on the processes involved, i.e. the stablity of CCS processes:

Definition 5. *(Stable processes (agents)) A process (or agent) is said to be* stable *if there's no τ-transition coming from it's root:*

$\vdash$ `STABLE` $E \iff \forall u\ E'.\ E$ `--`u`->` $E' \Rightarrow u \neq \tau$

Notice that, the stability of a CCS process doesn't imply the τ-free of all its sub-processes. Instead the definition only concerns on the first transition leading from the process (root).

Among other small lemmas, we have proved the following properties of weak bisimulation equivalence:

Proposition 7. *Properties of weak bisimulation equivalence)*

1. *Weak equivalence is substitutive under prefix operator:*

   $\vdash E \approx E' \Rightarrow \forall u.\ u..E \approx u..E'$ [WEAK_EQUIV_SUBST_PREFIX]

2. *Weak equivalence of stable agents is preserved by binary summation:*

   $\vdash E_1 \approx E_1' \wedge$ `STABLE` $E_1 \wedge$ `STABLE` $E_1' \wedge E_2 \approx E_2' \wedge$ `STABLE` $E_2 \wedge$ `STABLE` $E_2' \Rightarrow$
   E_1 `+` $E_2 \approx E_1'$ `+` E_2' [WEAK_EQUIV_PRESD_BY_SUM]

3. *Weak equivalence of stable agants is substitutive under binary summation on the right:*

   $\vdash E \approx E' \wedge$ `STABLE` $E \wedge$ `STABLE` $E' \Rightarrow \forall E''.\ E$ `+` $E'' \approx E'$ `+` E''[WEAK_EQUIV_SUBST_SUM_R]

4. *Weak equivalence of stable agants is substitutive under binary summation on the left:*

   $\vdash E \approx E' \wedge$ `STABLE` $E \wedge$ `STABLE` $E' \Rightarrow \forall E''.\ E''$ `+` $E \approx E''$ `+` E'[WEAK_EQUIV_SUBST_SUM_L]

5. *Weak equivalence is preserved by parallel operator:*

   $\vdash E_1 \approx E_1' \wedge E_2 \approx E_2' \Rightarrow E_1$ `||` $E_2 \approx E_1'$ `||` E_2' [WEAK_EQUIV_PRESD_BY_PAR]

6. *Weak equivalence is substitutive under restriction operator:*

   $\vdash E \approx E' \Rightarrow \forall L.\ \nu\ L\ E \approx \nu\ L\ E'$ [WEAK_EQUIV_SUBST_RESTR]

7. *Weak equivalence is substitutive under relabelling operator:*

   $\vdash E \approx E' \Rightarrow \forall rf.$ `relab` $E\ rf \approx$ `relab` $E'\ rf$ [WEAK_EQUIV_SUBST_RELAB]

Finally, we have proved that, strong equivalence implies weak equivalence:

Theorem 1. *(Strong equivalence implies weak equivalence)*

$\vdash E \sim E' \Rightarrow E \approx E'$ [STRONG_IMP_WEAK_EQUIV]

Here we omit all the algebraic laws for weak equivalence, because they were all easily derived from the corresponding algebraic laws for strong equivalence, except for the following τ-law:

Theorem 2. *The τ-law for weak equivalence)*

$\vdash \tau..E \approx E$ [TAU_WEAK]

6 Observation congruence and the algebraic laws

The concept of rooted weak bisimulation equivalence (also namsed *observation congruence*) is an "obvious fix" to convert weak bisimulation equivalence into a congruence. Its definition is not recursive but based on the definition of weak equivalence:

Definition 6. *(Observation congruence) Two CCS processes are observation congruence if and only if for any transition from one of them, there's a responding weak transition from the other, and the resulting two sub-processes are weak equivalence:*

$$\begin{array}{l}\vdash\ E\ \approx^c\ E' \iff \\ \quad \forall\, u\,. \\ \qquad (\forall\, E_1\,.\ E\ \texttt{--}u\texttt{->}\ E_1\ \Rightarrow\ \exists\, E_2\,.\ E'\ \texttt{==}u\texttt{=>>}\ E_2\ \wedge\ E_1\ \approx\ E_2)\ \wedge \\ \qquad \forall\, E_2\,.\ E'\ \texttt{--}u\texttt{->}\ E_2\ \Rightarrow\ \exists\, E_1\,.\ E\ \texttt{==}u\texttt{=>>}\ E_1\ \wedge\ E_1\ \approx\ E_2 \end{array} \qquad \texttt{[OBS\_CONGR]}$$

By observing the differences between the definition of observation equivalence (weak equivalence) and congruence, we can see that, observation equivalence requires a little more: for each τ-transition from one process, the other process must response with at least one τ-transition. Thus what's immediately proven is the following two theorems:

Theorem 3. *(Observation congruence implies observation equivalence)*

$$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ E\ \approx\ E' \qquad \texttt{[OBS\_CONGR\_IMP\_WEAK\_EQUIV]}$$

Theorem 4. *(Observation equivalence on stable agents implies observation congruence)*

$$\vdash\ E\ \approx\ E'\ \wedge\ \texttt{STABLE}\ E\ \wedge\ \texttt{STABLE}\ E'\ \Rightarrow\ E\ \approx^c\ E' \qquad \texttt{[WEAK\_EQUIV\_STABLE\_IMP\_CONGR]}$$

Surprisingly, it's not trivial to prove that, the observation equivalence is indeed an equivalence relation. The reflexitivy and symmetry are trivial:

Proposition 8. *(The reflexitivy and symmetry of observation congruence)*

$$\vdash\ E\ \approx^c\ E \qquad \texttt{[OBS\_CONGR\_REFL]}$$
$$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ E'\ \approx^c\ E \qquad \texttt{[OBS\_CONGR\_SYM]}$$

But the transitivity is hard to prove.[7] Our proof here is based on the following lemmas:

Lemma 1. *If two processes E and E' are observation congruence, then for any EPS transition coming from E, there's a corresponding EPS transition from E', and the resulting two subprocesses are weakly equivalent:*

$$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ \forall\, E_1\,.\ E\ \stackrel{\epsilon}{\Rightarrow}\ E_1\ \Rightarrow\ \exists\, E_2\,.\ E'\ \stackrel{\epsilon}{\Rightarrow}\ E_2\ \wedge\ E_1\ \approx\ E_2 \qquad \texttt{[OBS\_CONGR\_EPS]}$$

Proof. By (right) induction[8] on the number of τ in the EPS transition of E. In the base case, there's no τ at all, the E transites to itself. And in this case E' can respond with itself, which is also an EPS transition:

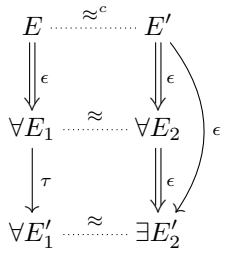
$$\begin{array}{ccc} E & \overset{\approx^c}{\cdots\cdots} & E' \\ \Big| = & & \Big| = \\ E & \overset{\approx}{\cdots\cdots} & E' \end{array}$$

For the induction case, suppose the proposition is true for zero or more τ transitions except for the last step, that's, $\forall E, \exists E1, E2$, such that $E\ \stackrel{\epsilon}{\Rightarrow}\ E_1$, $E'\ \stackrel{\epsilon}{\Rightarrow}\ E_2$ and $E_1\ \approx\ E_2$. Now by definition of weak equivalence, if $E_1\ \texttt{-}\tau\texttt{->}\ E_1'$ then there exists $E2'$ such that $E_2\ \stackrel{\epsilon}{\Rightarrow}\ E_2'$ and $E_1'\ \approx\ E_2'$. Then

[7] Actually it's not proven in the old work, the formal proofs that we did in this project is completely new.

[8] The induction theorem used here is `EPS_ind_right`.

by transitivity of EPS, we have $E' \stackrel{\epsilon}{\Rightarrow} E_2 \wedge E_2 \stackrel{\epsilon}{\Rightarrow} E_2' \Rightarrow E' \stackrel{\epsilon}{\Rightarrow} E_2'$, thus E_2' is a valid response required by observation congruence:

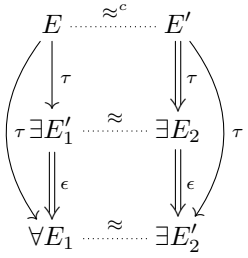


□

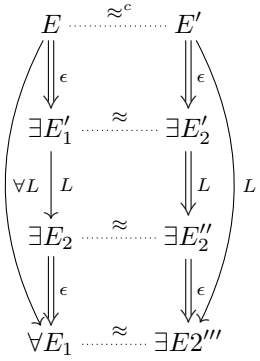
Lemma 2. *If two processes E and E' are observation congruence, then for any weak transition coming from E, there's a corresponding weak transition from E', and the resulting two subprocesses are weakly equivalent:*

$\vdash E \approx^c E' \Rightarrow \forall u\ E_1.\ E\ \texttt{==}u\texttt{=>>}\ E_1 \Rightarrow \exists E_2.\ E'\ \texttt{==}u\texttt{=>>}\ E_2 \wedge E_1 \approx E_2$

Proof. (sketch Consider the two cases when the action is τ or not τ. For all weak τ-transitions coming from E, the observation congruence requires that there's at least one τ following E' and the resulting two sub-processes, say E_1' and E_2 are weak equivalence. Then the desired responses can be found by using a similar existence lemma for weak equivalence:



For all the non-τ weak transitions from E, the proof follows from previous lemma and a similar existence lemma for weak equivalence. The following figure is a sketch for the proof of this case:



In the previous figure, the existence of E_2' follows by previous lemma, the existence of E_2'' follows by the definition of weak equivalence, and the existence of E_2''' follows by the next existence lemma of weak equivalence. □

The existence lemma for weak equivalences that we mentioned in previous proof is the following one:

Lemma 3. $\vdash E \stackrel{\epsilon}{\Rightarrow} E_1 \Rightarrow$
$\forall\ Wbsm\ E'.$
$\texttt{WEAK\_BISIM}\ Wbsm \wedge Wbsm\ E\ E' \Rightarrow \exists E_2.\ E' \stackrel{\epsilon}{\Rightarrow} E_2 \wedge Wbsm\ E_1\ E_2$

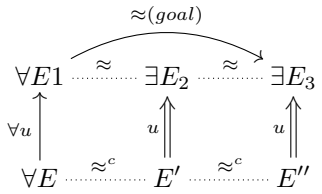
Now we prove the transitivity of observation congruence:

Theorem 5. *(Transitivity of Observation Congruence)*

$$\vdash E \approx^c E' \wedge E' \approx^c E'' \Rightarrow E \approx^c E'' \qquad \texttt{[OBS\_CONGR\_TRANS]}$$

Proof. Suppose $E \approx^c E'$ and $E' \approx^c E''$, we're going to prove $E \approx^c E''$ by checking directly the definition of observation congruence.

For any u and E_1 which satisfy E `-`u`->` E_1, by definition of observation congruence, there exists E_2 such that E' `==`u`=>` E_2 with $E_1 \approx E_2$. By above Lemma 2, there exists another E_3 such that E'' `==`u`=>` E_3 with $E_2 \approx E_3$. By the already proven transitivity of weak equivalence, $E_1 \approx E_3$, thus E_3 is the required process which saitisfies the definition of observation congruence. This proves the first part. The other part is completely symmetric.



□

Then we have proved the substitutivity of observation congruence under various CCS process operators:

Proposition 9. *1. Observation congruence is substitutive under the prefix operator:*

$$\vdash E \approx^c E' \Rightarrow \forall u.\ u..E \approx^c u..E' \qquad \texttt{[OBS\_CONGR\_SUBST\_PREFIX]}$$

2. Observation congruence is substitutive under binary summation:

$$\vdash p \approx^c q \wedge r \approx^c s \Rightarrow p + r \approx^c q + s \qquad \texttt{[OBS\_CONGR\_PRESD\_BY\_SUM]}$$

3. Observation congruence is preserved by parallel composition:

$$\vdash E_1 \approx^c E_1' \wedge E_2 \approx^c E_2' \Rightarrow E_1 \texttt{ || } E_2 \approx^c E_1' \texttt{ || } E_2' \qquad \texttt{[OBS\_CONGR\_PRESD\_BY\_PAR]}$$

4. Observation congruence is substitutive under the restriction operator:

$$\vdash E \approx^c E' \Rightarrow \forall L.\ \nu\ L\ E \approx^c \nu\ L\ E' \qquad \texttt{[OBS\_CONGR\_SUBST\_RESTR]}$$

5. Observation congruence is substitutive under the relabeling operator:

$$\vdash E \approx^c E' \Rightarrow \forall rf.\ \texttt{relab}\ E\ rf \approx^c \texttt{relab}\ E'\ rf \qquad \texttt{[OBS\_CONGR\_SUBST\_RELAB]}$$

Finally, like the case for weak equivalence, we can easily prove the relationship between strong equivalence and observation congruence:

Theorem 6. *(Strong equivalence implies observation congruence)*

$$\vdash E \sim E' \Rightarrow E \approx^c E' \qquad \texttt{[STRONG\_IMP\_OBS\_CONGR]}$$

With this result, all algebraic laws for observation congruence can be derived from the corresponding algebraic laws of strong equivalence. Here we omit these theorems, except for the following four τ-laws:

Theorem 7. *(The τ-laws for observation congruence)*

$$\vdash u..\tau..E \approx^c u..E \qquad \texttt{[TAU1]}$$
$$\vdash E + \tau..E \approx^c \tau..E \qquad \texttt{[TAU2]}$$
$$\vdash u..(E + \tau..E') + u..E' \approx^c u..(E + \tau..E') \qquad \texttt{[TAU3]}$$
$$\vdash E + \tau..(E' + E) \approx^c \tau..(E' + E) \qquad \texttt{[TAU\_STRAT]}$$

7 Deng lemma and Hennessy lemma

The relationship between weak equivalence and observation congruence was an interesting research topic, and there're many deep lemmas related. In this project, we have proved two such deep lemmas. The first one is the following Deng Lemma (for weak bisimularity[9]):

Theorem 8. *(Deng lemma for weak bisimilarity) If $p \approx q$, then one of the following three cases holds:*

1. *$\exists p'$ such that p -τ-> p' and $p' \approx q$, or*
2. *$\exists q'$ such that q -τ-> q' and $p \approx q'$, or*
3. *$p \approx^c q$.*

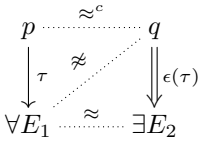
$$\vdash p \approx q \Rightarrow (\exists p'.\ p \text{ --}\tau\text{->} p' \wedge p' \approx q) \vee (\exists q'.\ q \text{ --}\tau\text{->} q' \wedge p \approx q') \vee p \approx^c q \qquad \texttt{[DENG\_LEMMA]}$$

Proof. Actually there's no need to consider thee difference cases. Using the logical tautology $(\neg P \wedge \neg Q \Rightarrow R) \Rightarrow P \vee Q \vee R$, the theorem can be reduced to the following goal:

Prove $p \approx^c q$, with the following three assumptions:
1. $p \approx q$
2. $\neg\exists p'.\ p$ -τ-> $p' \wedge p' \approx q$
3. $\neg\exists q'.\ q$ -τ-> $q' \wedge p \approx q'$

Now we check the definition of observation congruence: for any transition from p, say p -u-> E_1, consider the cases when $u = \tau$ and $u \neq \tau$:

1. If $u = \tau$, then by $p \approx q$ and the definition of weak equivalence, there exists E_2 such that $q \stackrel{\varsigma}{\Rightarrow} E_2$ and $E_1 \approx E_2$. But by assumption we know $q \neq E2$, thus $q \stackrel{\varsigma}{\Rightarrow} E_2$ contains at least one τ-transition, thus is actually q ==τ=» E_2, which is required by the definition of observation congruence for $p \approx q$.



2. If $u = L$, then the requirement of observation congruence is directly satisfied.

The other direction is completely symmetric. □

Now we start to prove Hennessy Lemma:

Theorem 9. *(Hennessy Lemma) For any processes p and q, $p \approx q$ if and only if ($p \approx^c q$ or $p \approx^c \tau..q$ or $\tau..p \approx^c q$):*

$$\vdash p \approx q \iff p \approx^c q \vee p \approx^c \tau..q \vee \tau..p \approx^c q \qquad \texttt{[HENNESSY\_LEMMA]}$$

Proof. The "if" part (from right to left) can be easily derived by applying `OBS_CONGR_IMP_WEAK_EQUIV`, `TAU_WEAK`, `WEAK_EQUIV_SYM` and `WEAK_EQUIV_TRANS`. We'll focus on the hard "only if" part (from left to right). The proof represent here is slightly simplier than the one in [5], but the idea is the same. The proof is based on creative case analysis.

If there exists an E such that p -τ-> $E \wedge E \approx q$ than we can prove that $p \approx^c \tau..q$ by expanding $p \approx q$ by `OBS_PROPERTY_STAR`. The other needed theorems are the definition of weak transition, `EPS_REFL`, SOS rule `PREFIX` and `TRANS_PREFIX`, `TAU_PREFIX_WEAK_TRANS` and `TRANS_IMP_WEAK_TRANS`.

If there's no E such that p -τ-> $E \wedge E \approx q$, we can further check if there exist an E such that q -τ-> $E \wedge p \approx E$, and in this case we can prove $\tau..p \approx^c q$ in the same way as the above case.

Otherwise we got exactly the same condition as in Deng Lemma (after the initial goal reduced in the previous proof), and in this case we can directly prove that $p \approx^c q$.

[9] The original Deng lemma is for another kind of equivalence relation called *rooted branching bisimularity*, which is not touched in this project.

The purpose of this formal proof has basically shown that, for most informal proofs in Concurrency Theory which doesn't depend on external mathmatics theories, the author has got the ability to express it in HOL theorem prover.

8 Coarsest congruence contained in weak equivalence

In this section, we present two formal proofs for the "coarsest congruence contained in weak equivalence", namely:

Proposition 10. *(Coarsest congruence contained in ≈) For any processes p and q, $p \approx^c q$ if and only if $\forall r.\ p + r \approx q + r$.*

But first of all, we need to explain how the theorem name comes and why it concerns only on the summations at the right side of the statement. We get these knowledge from [7]. The next several definitions and propositions are only informally defined and proved:

Definition 7. *(Congruence) An equivalence relation ≈[10] on a specific space of CCS processes is a* congruence *iff for every n-ary operator f, one has $g_1 \approx h_1 \wedge \cdots g_n \approx h_n \Rightarrow f(g_1, \ldots, g_n) \approx f(h_1, \ldots, h_n)$. This is the case iff for every semantic context $C[\cdot]$ on has $g \approx h \Rightarrow C[g] \approx C[h]$.*

Definition 8. *(Constructing congruences from equivalence relation) Given an equivalence relation ∼[11], define $\approx^c$ by*

$$g \sim^c h \text{ iff } C[g] \sim C[h] \text{ for every semantic context } C[\cdot] \tag{1}$$

Proposition 11. *$\sim^c$ is a congruence.*

Proof. By construction, $\sim^c$ is a congruence. For if $g \sim^c h$ and $D[\cdot]$ is a semantic context, then for every semantic context $C[\cdot]$ also $C[D[\cdot]]$ is a semantic context, so $\forall C[\cdot].(C[D[g]] \sim C[D[h]])$ and hence $D[g] \sim^c D[h]$. □

Proposition 12. *$\sim^c$ is finer than ∼.*

Proof. The trivial context guarantees that $g \sim^c h \Rightarrow g \sim h$, so $\sim^c$ is finer than ∼. □

Proposition 13. *$\sim^c$ is the coarsest congruence finer than ∼.*

Proof. If ≈ is any congruence finer than ∼, then

$$g \approx h \Rightarrow \forall C[\cdot].(C[g] \approx C[h]) \Rightarrow \forall C[\cdot].(C[g] \sim C[h]) \Rightarrow g \sim^c h. \tag{2}$$

Thus ≈ is finer than $\sim^c$. (i.e. $\sim^c$ is coarser than ≈, then the arbitrarity of ≈ implies that $\sim^c$ is coarsest.) □

We know that weak equivalence is not congruence because the weak equivalence doesn't hold on summation for any processes, that is[12],

$$\forall p, q. \quad p + r \sim_w q + r \text{ doesn't hold for all } r. \tag{3}$$

Since $\sim_w$ is not congruence, it's natural to try to construct an congruence $\sim_w^c$ from $\sim_w$ by definition,

$$g \sim_w^c h \text{ iff } C[g] \sim_w C[h] \text{ for every semantic context } C[\cdot]. \tag{4}$$

But we know the substitutivity of weak equivalence already holds for every CCS operator except for the summation. Thus above equation (and definition) can be simplified as:

[10] The symbol ≈ here shouldn't be understood as weak equivalence.
[11] The Symbol ∼ here shouldn't be understood as strong equivalence.
[12] Here we temporarily use the symbol $\sim_w$ for weak equivalence, to prevent the confliction with the symbols used in above several definitions in the current section.

Definition 9. *(Weak bisimulation congruence) The coarasest congruence w.r.t the summation (+) that is finer then weak bisimulation equivalence is called* weak bisimulation congruence*, notation* $\sim_w^c$*:*

$$g \sim_w^c h \text{ iff } g + r \sim_w h + r \text{ for every process } r. \tag{5}$$

So far, the weak bisimulation congruence $\sim_w^c$ defined above is irrelevant with rooted weak bisimulation (a.k.a. observation congruence) $\approx^c$, which has the following standard definition also based on weak equivalence:

$$\begin{array}{l} \vdash E \approx^c E' \iff \\ \quad \forall u. \\ \quad\quad (\forall E_1.\ E \texttt{ --}u\texttt{-> } E_1 \Rightarrow \exists E_2.\ E' \texttt{ ==}u\texttt{=>> } E_2 \wedge E_1 \approx E_2) \wedge \\ \quad\quad \forall E_2.\ E' \texttt{ --}u\texttt{-> } E_2 \Rightarrow \exists E_1.\ E \texttt{ ==}u\texttt{=>> } E_1 \wedge E_1 \approx E_2 \end{array}$$

One of the most interesting result in concurrency theory is the theorem saying that $\approx^c$ actually coincides with $\sim_w^c$, i.e. $\approx^c = \sim_w^c$. If we eliminate the definition of $\sim_w^c$ from this equation, finally we get the initial proposition at the beginning of this section. Now its name explains itself.

It's trivial to prove that $\approx^c$ is finer than $\sim_w^c$ ($\approx$) (from left to right), because we have already proved that $\approx^c$ is congruence and $\approx^c$ implies $\approx$ (`OBS_CONGR_IMP_WEAK_EQUIV`):

Theorem 10. *(The $\Rightarrow$ direction of "Coarsest congruence contained in $\approx$")*

$$\vdash p \approx^c q \Rightarrow \forall r.\ p \texttt{ + } r \approx q \texttt{ + } r \qquad \texttt{[COARSEST\_CONGR\_LR]}$$

The other side is very hard to prove, especially when there's no restriction on the processes. A classic restriction is to assume that, the two processes didn't use up all possible labels. In [6] (Proposition 3 in Chapter 7, p. 153), Robin Milner simply called this theorem "Proposition 3":

Proposition 14. *(Proposition 3 of observation congruence) Assume that $\mathcal{L}(P) \cup \mathcal{L}(Q) \neq \mathcal{L}$. Then $P \approx^c Q$ iff, for all R, $P + R \approx Q + R$.*

And in [5] (Theorem 4.5 in Chapter 4, p. 185), Prof. Gorrieri has called it "Coarsest congruence contained in $\approx$" (so did us in this paper):

Theorem 11. *(Coarsest congruence contained in $\approx$) Assume that* $\mathrm{fn}(p) \cup \mathrm{fn}(q) \neq \mathscr{L}$*. Then $p \approx^c q$ if and only if $p + r \approx q + r$ for all $r \in \mathscr{P}$.*

Both $\mathcal{L}(\cdot)$ and $\mathrm{fn}(\cdot)$ used in above theorems mean the set of "non-τ actions" (i.e. labels) used in a given process.

We analysized the proof of abvoe theorem and have found that, the assumption that the two processes didn't use up all available labels, although concise, can be greatly weakened. In the proof of any theorem, if additional assumptions must be added, that's becase the proof cannot be finished without such assumptions. Using theorem prover, we firstly try to prove the theorem using exactly the same method but without the assumptions, and then supply minimal version of the original assumptions for bypassing the proof. As a result, in this project we have proved the following stronger version of the "Coarsest congruence contained in $\approx$" theorem, the following proved theorem represents the hard part:

Definition 10. *(Processes having free actions) A CCS process is said to have* free actions *if there exists an non-τ action such that it doesn't appear in any transition or weak transition directly leading from the root of the process:*

$$\vdash \texttt{free\_action } p \iff \exists a.\ \forall p'.\ \neg(p \texttt{ ==label } a\texttt{=>> } p')$$

Theorem 12. *(Stronger version of "Coarsest congruence contained in $\approx$", only the hard part) Assuming for two processes p and q have free actions, then $p \approx^c q$ if $p + r \approx q + r$ for all $r \in \mathscr{P}$:*

$$\vdash \texttt{free\_action } p \wedge \texttt{free\_action } q \Rightarrow (\forall r.\ p \texttt{ + } r \approx q \texttt{ + } r) \Rightarrow p \approx^c q \texttt{[COARSEST\_CONGR\_RL]}$$

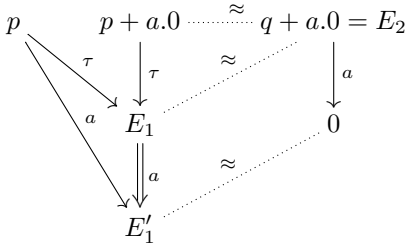
The assumptions are weaken because, even p and q may have used all possible actions in their transition graphs, as long as there's one such free action for their first-step weak transitions, therefore the theorem still holds. Also noticed that, the two processes do not have to share the same free actions, this property focuses on single process.

It's also worth to explain the reason why these assumptions are not automatically satisfied, given the fact that our CCS datatype is finitary, i.e. there's no infinite sums or parallels. The reason is simple, our CCS datatype has been generalized, and *we may not have infinite actions to use*. In the extreme case, there may be only three available actions (τ, a and $\overline{a}$), and if p and q contain transitions for all these actions, then we actually cannot say the "coarsest congruence contained in $\approx$" coincides with observation congruence by applying above theorem.

Proof. (Proof of the stronger version of "Coarsest congruence contained in $\approx$") The kernel idea in this proof is to use that free action, say a, and have $p + a.0 \approx q + a.0$ as the working basis. Then for any transition from $p + a.0$, say $p + a.0 \xRightarrow{u} E_1$, there must be a weak transition of the same action u (or EPS when $u = \tau$) coming from $q + a.0$ as the response. We're going to use the free-action assumptions to conclude that, when $u = \tau$, that EPS must contain at least one τ (thus satisfied the definition of observation congruence):

$$\begin{array}{ccc} p + a.0 & \overset{\approx}{\cdots\cdots} & q + a.0 \\ \Big\downarrow{\scriptstyle u=\tau} & & \Big\Downarrow{\scriptstyle \epsilon} \\ E_1 & \overset{\approx}{\cdots\cdots} & E_2 \end{array}$$

Indeed, if the EPS leading from $q+a.0$ actually contains no τ-transition, that is, $q+a.0 = E_2$, then E_1 and E_2 cannot be weak equivalence: for any a-transition from $q + a.0$, $E1$ must response with a weak a-transition as $E_1 \xRightarrow{a} E_1'$, but this means $p \xRightarrow{a} E_1'$, which is impossible by free-action assumption on p:



Once we have $q + a.0 \xRightarrow{\tau} E2$, the first τ-transition must comes from q, then it's obvious to see that E_2 is a valid response required by observation congruence of p and q in this case.

When $p \xrightarrow{L} E_1$, we have $p + a.0 \xrightarrow{L} E_1$, then there's an E_2 such that $q + a.0 \xRightarrow{L} E_2$. We can further conclude that $q \xRightarrow{L} E_2$ because by free-action assumption $L \neq a$. This finishes the first half of the proof, the second half (for all transition coming from q) is completely symmetric. □

Combining the easy and hard parts, the following theorem is proved:

Theorem 13. *(Coarsest congruence contained in $\approx$)*

$\vdash$ `free_action` $p \wedge$ `free_action` $q \Rightarrow (p \approx^c q \iff \forall r.\ p$ `+` $r \approx q$ `+` $r)$

8.1 Without cardinality assumptions

In 2005, Rob J. van Glabbeek published a paper [7] showing that "the weak bisimulation congruence can be characterised as rooted weak bisimulation equivalence, even without making assumptions on the cardinality of the sets of states or actions of the process under consideration". That is to say, above "Coarsest congruence contained in $\approx$" theorem holds even for two arbitrary processes! The idea is actually from Jan Willem Klop back to the 80s, but it's not published until that 2005 paper. *We carefully investigated this paper and focused on the formalization of the proof contained in the paper, with all remain plans of this "tirocinio" project cancelled.* It turns out that, the proof

represented in that paper is very very interesting and went beyond usual mathematic tools needed in Concurrency Theory.

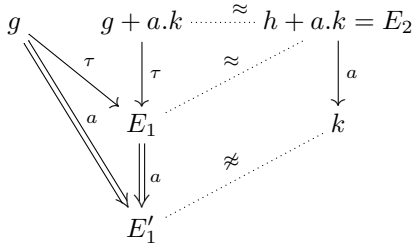
The main result is the following version of the hard part of "Coarsest congruence contained in $\approx$" theorem under new assumptions:

Theorem 14. *(Coarsest congruence contained in $\approx$, new assumptions) For any two CCS processes g and h, if there exists another stable (i.e. first-step transitions are never τ-transition) process k which is not weak bisimlar with any sub-process follows from p and q by* one-step *weak transitions, then $g \approx^c h$ if $g + r \approx h + r$ for all $r \in \mathscr{P}$.*

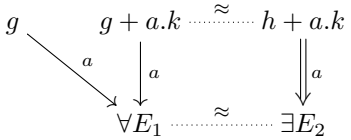
$$
\begin{array}{l}
\vdash (\exists\, k. \\
\quad \texttt{STABLE}\ k \wedge (\forall\, g'\ u.\ g\ \texttt{==}u\texttt{=>>}\ g' \Rightarrow \neg(g' \approx k)) \wedge \\
\quad \forall\, h'\ u.\ h\ \texttt{==}u\texttt{=>>}\ h' \Rightarrow \neg(h' \approx k)) \Rightarrow \\
(\forall\, r.\ g\ \texttt{+}\ r \approx h\ \texttt{+}\ r) \Rightarrow \\
g \approx^c h
\end{array}
$$

Proof. Assuming the existence of that special process k, and take an arbitrary non-τ action, say a (this is always possible in our setting, because in higher order logic any valid type must contain at least one value), we'll use the fact that $g + a.k \approx h + a.k$ as our working basis. For all transitions from g, say $g \xrightarrow{u} E_1$, we're going to prove that, there must be a corresponding weak transition such that $h \stackrel{u}{\Longrightarrow} E_2$, and $E_1 \approx E_2$ (thus $g \approx^c h$. There're three cases to consider:

1. τ-transitions: $g \xrightarrow{\tau} E_1$. By SOS rule (Sum_1), we have $g + a.k \xrightarrow{\tau} E_1$, now by $g + a.k \approx h + a.k$ and the property (*) of weak equivalence, there exists an E_2 such that $h + a.k \stackrel{\epsilon}{\Longrightarrow} E_2$. We can use the property of k to assert that, such an EPS transition must contains at least one τ-transition. Because if it's not, then $h + a.k = E_2$, and since $E_1 \approx E_2$, for transition $h + a.k \xrightarrow{a} k$, E_1 must make a response by $E_1 \stackrel{a}{\Longrightarrow} E_1'$, and as the result we have $g \stackrel{a}{\Longrightarrow} E_1'$ and $E_1' \approx k$, which is impossible by the special choice of k:



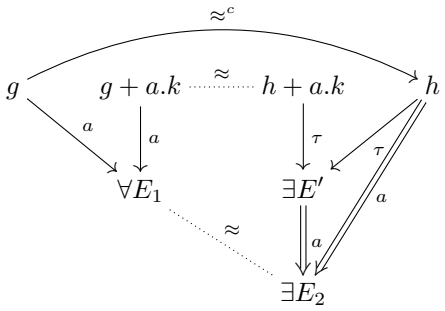
2. If there's a a-transition coming from g (means that the arbitrary chosen action a is normally used by processes g and h), that is, $g \xrightarrow{a} E_1$, also $g + a.k \xrightarrow{a} E_1$, by property (*) of weak equivalence, there exists E_2 such that $h + a.k \stackrel{a}{\Longrightarrow} E_2$:



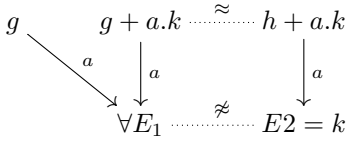
   We must further divide this weak transition into two cases based on its first step:

   (a) If the first step is a τ-transition, then for sure this entire weak transition must come from h (otherwise the first step would be an a-transition from $a.k$). And in this case we can

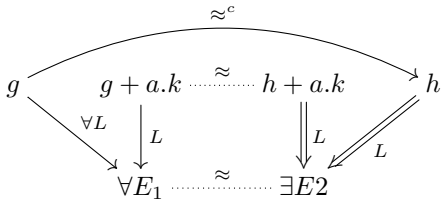
easily conclude $h \stackrel{a}{\Longrightarrow} E_2$ without using the property of k:



(b) If the first step is an a-transition, we can prove that, this a-transition must come from h (then the proof finishes for the entire a-transition case). Because if it's from the $a.k$, since k is stable, then there's no other coice but $E_2 = k$ and $E_1 \approx E_2$. This is again impossible for the special choice of k:



3. For other L-transitions coming from g, where $L \neq a$ and $L \neq \tau$. As a response to $g+a.k \xrightarrow{L} E_1$, we have $h + a.k \stackrel{L}{\Longrightarrow} E_2$ and $E_1 \approx E_2$. It's obvious that $h \stackrel{L}{\Longrightarrow} E_2$ in this case, no matter what the first step is (it can only be τ and L) and this satisfies the requirement of observation congruence natually:



The other direction (for all transitions coming from h) is completely symmetric. Combining all the cases, we have $g \approx^c h$. □

Now it remains to prove the assumption in above theorem always exists.

8.2 Arbitrary Many Non-bisimilar Processes

Strong equivalence, weak equivalence, observation congruence, they're all equivalance relations on CCS process space. General speaking, each equivalence relation must have *partitioned* all processes into several disjoint equivalance classes: processes in the same equivalance class are equivalent, and processes in different equivalence class are not equivalent.

The assumption in previous Theorem 14 requires the existence of a special CCS process, which is not weak equivalence to any sub-process leading from the two root processes by weak transitions. On worst cases, there may be infinite such sub-processes[13] Thus there's no essential differences to consider all states in the process group instead.

Then it's natural to consider, if there exists infinite equivalence classes of CCS processes, then it should be possible to choose one which is not equivalent with all states in the graphs of the two given processes. It turns out that, after Jan Willem Klop, there exists such processes, in which each of them forms a new equivalence class, we call them "Klop processes" in this paper:

[13] Even the CCS is finite branching, that's because after a weak transition, the end process may have an infinite τ-chain, and with each τ-transition added into the weak transition, the new end process is still a valid weak transition, thus lead to infinite number of weak transitions.

Definition 11. *(Klop processes) For each ordinal λ, and an arbitrary chosen non-τ action a, define a CCS process k_λ as follows:*

1. $k_0 = 0$,
2. $k_{\lambda+1} = k_\lambda + a.k_\lambda$ *and*
3. *for λ a limit ordinal, $k_\lambda = \sum_{\mu<\lambda} k_\mu$, meaning that k_λ is constructed from all graphs k_μ for $\mu < \lambda$ by identifying their root.*

Unfortunately, it's impossible to express infinite sums in our CCS datatype settings[14], nor possible to define recursive function based on ordinals in HOL4, partly because the ordinalTheory [8] and cardinalTheory are not part of the core theories in HOL4. Therefore we have followed a two-step approach in this project: first we consider only the finite-state CCS, then we proceed with the general case.

If both processes p and q are finite-state CCS processes, that is, the number of reachable states from p and q are both finite. And in this case, the following limited version of Klop processes can be defined as a recursive function (on natural numbers) in HOL4:

Definition 12. *(Klop processes as recursive function on natural numbers)*

```
KLOP a 0 = nil
KLOP a (SUC n) = KLOP a n + label a..KLOP a n                    [KLOP_def]
```

By induction on the definition of Klop processes and SOS inference rules (Sum_1) and (Sum_2), we can easily prove the following properties of KLOP functions:

Proposition 15. *(Properties of Klop functions and processes)*

1. *All Klop processes are stable:*

   $\vdash$ `STABLE (KLOP a n)` `[KLOP_PROP0]`

2. *All transitions of a Klop process must lead to another smaller Klop process, and any smaller Klop process must be a possible transition of a larger Klop process:*

   $\vdash$ `KLOP a n --label a->` $E \iff \exists m.\ m < n \wedge E =$ `KLOP a m` `[KLOP_PROP1]`

3. *The weak transition version of above property:*

   $\vdash$ `KLOP a n ==label a=>>` $E \iff \exists m.\ m < n \wedge E =$ `KLOP a m` `[KLOP_PROP1_WK]`

4. *All Klop processes are distinct according to strong equivalence:*

   $\vdash m < n \Rightarrow \neg$(`KLOP a m` $\sim$ `KLOP a n`) `[KLOP_PROP2]`

5. *All Klop processes are distinct according to weak equivalence:*

   $\vdash m < n \Rightarrow \neg$(`KLOP a m` $\approx$ `KLOP a n`) `[KLOP_PROP2_WK]`

6. *Klop functions are one-one:*

   $\vdash$ `ONE_ONE (KLOP a)` `KLOP_ONE_ONE`

Once we have a recursive function defined on all natural numbers $0, 1, \ldots$, we can construct a set containing all these Klop processes, and the set is countable infinite. On the other side, the number all states from two finite-state CCS processes p and q is finite. Choosing from an infinite set for an element distinct with any subprocess leading from p and q, is always possible. This result is purely mathematical, completely falling into basic set theory:

Lemma 4. *Given an equivalence relation R defined on a type, and two sets A, B of elements in this type, A is finite, B is infinite, and all elements in B are not equivalent, then there exists an element k in B which is not equivalent with any element in A:*

[14] And such infinite sums seems to go beyond the ability of the HOL's Datatype package

$\vdash$ `equivalence` $R \Rightarrow$
`FINITE` $A \wedge$ `INFINITE` $B \wedge$
$(\forall x\ y.\ x \in B \wedge y \in B \wedge x \neq y \Rightarrow \neg R\ x\ y) \Rightarrow$
$\exists k.\ k \in B \wedge \forall n.\ n \in A \Rightarrow \neg R\ n\ k$

Proof. We built an explicit mapping f from A to B[15], for all $x \in A$, $y = f(x)$ if $y \in B$ and y is equivalent with x. But it's possible that no element in B is equivalent with x, and in this case we just choose an arbitrary element as $f(x)$. Such a mapping is to make sure the range of f always fall into B.

Now we can map A to a subset of B, say B_0, and the cardinality of B_0 must be equal or smaller than the cardinality of A, thus finite. Now we choose an element k from the rest part of B, this element is the desire one, because for any element $xinA$, if it's equivalent with k, consider two cases for $y = f(x) \in B_0$:

1. y is equivalent with x. In this case by transitivity of R, we have two distinct elements y and k, one in B_0, the other in $B \setminus B_0$, they're equivalent. This violats the assumption that all elements in B are distinct.
2. y is arbitrary chosen because there's no equivalent element for x in B. But we already know one: k.

Thus there's no element in x which is equivalent with k. □

To reason about finite-state CCS, we also need to define the concept of "finite-state":

Definition 13. *(Definitions related to finite-state CCS)*

1. *Define* reachable *as the RTC of a relation, which indicates the existence of a transition between two processes:*

   $\vdash$ `Reachable` $= (\lambda E\ E'.\ \exists u.\ E$ `--`u`->` $E')^*$

2. *The "nodes" of a process is the set of all processes reachable from it:*

   $\vdash$ `NODES` $p = \{q$ `|` `Reachable` $p\ q\}$

3. *A process is finite-state if the set of nodes is finite:*

   $\vdash$ `FINITE_STATE` $p \iff$ `FINITE (NODES` p`)`

Among many properties of above definitions, we mainly rely on the following "obvious" property on weak transitions:

Proposition 16. *If p weakly transit to q, then q must be in the node set of p:*

$\vdash p$ `==`u`=>>` $q \Rightarrow q \in$ `NODES` p `[WEAK_TRANS_IN_NODES]`

Using all above results, now we can easily prove the following finite version of "Klop lemma":

Lemma 5. *Klop lemma, the finite version For any two finite-state CCS p and q, there exists another process k, which is not weak equivalent with any sub-process weakly transited from p and q:*

$\vdash \forall g\ h.$
`FINITE_STATE` $g \wedge$ `FINITE_STATE` $h \Rightarrow$
$\exists k.$
`STABLE` $k \wedge (\forall g'\ u.\ g$ `==`u`=>>` $g' \Rightarrow \neg(g' \approx k)) \wedge$
$\forall h'\ u.\ h$ `==`u`=>>` $h' \Rightarrow \neg(h' \approx k)$ `[KLOP_LEMMA_FINITE]`

Combining above lemma, Theorem 14 and Theorem 10, we can easily prove the following theorem for finite-state CCS:

Theorem 15. *(Coarsest congruence contained in $\approx$ for finite-state CCS)*

$\vdash$ `FINITE_STATE` $g \wedge$ `FINITE_STATE` $h \Rightarrow$
$(g \approx^c h \iff \forall r.\ g$ `+` $r \approx h$ `+` $r)$

[15] There're multiple ways to prove this lemma, a simpler proof is to make a reverse mapping from B to the power set of A (or further use the Axiom of Choice (AC) to make a mapping from B to A), then the non-injectivity of this mapping will contradict the fact that all elements in the infinite set are distinct. Our proof doesn't need AC, and it relies on very simple truths about sets.

8.3 Full version

Now we turn to the general case. The number of nodes in the graph of a CCS process may be infinite, and in worst case such an "infinite" may be uncountable or even larger. In such cases, it's not guaranteed to find a Klop process $K_n, n \in \mathbb{N}$ which is not weak equivalence with any node (sub-process) in the graph. To formalize such a proof, we have to use ordinals instead of natural numbers in the definition of Klop processes.

Unfortunately, ordinals are not part of HOL's core theories, and currently it's impossible to define recursive functions on ordinals as the conservative extension of the HOL logic. Further more, our CCS datatype is inductively defined, there's no way to have infinite sum of CCS processes, not to mention an arbitrary large (at least uncountable) set of CCS summations. Due to these limitations, we have use an *axiom* to represent the Klop processes defined on ordinals, and instead of trying to represent infintie sums, we turn to focus on the folloing SOS infinite sum rule of CCS processes:

$$\text{(Summ)}\ \frac{p_i \xrightarrow{\mu} p'}{\sum_{i \in I} p_i \xrightarrow{\mu} p'}\ i \in I$$

The idea is, for any process p_i which has a transition to p', after adding to a possible infinite number of other processes, it still has the transition to p'.

Based on above idea, we have defined a new constant `Klop` (in captical cases, to be distinct with previous finite version `KLOP`') in HOL Logic and then defined the following properties for this constant:

```
Klop a 0o = nil
Klop a n⁺ --u-> E ⟺
u = label a ∧ E = Klop a n ∨ Klop a n --u-> E
islimit n ⇒ (Klop a n --u-> E ⟺ ∃m. m < n ∧ Klop a m --u-> E)
```

This definition focus on the possible transitions to Klop processes, and it's not hard to imagine the equivalence with the Math definition in the original paper. Usually it's dangerous to use axioms (HOL's command `new_axiom`, because it may introduce inconsistencies into HOL Logic and as a result all false theorems could be provable. But here we only used such an axiom to define a recursive function, and what's left unproved is just the fact that such a definition does terminate on every ordinals. In the future, once we have found a solution to remove the use of `new_axiom` but still have exactly the same transition behavior for the Klop processes, the rest theorems and proofs in this paper will be still valid without essential modifications.

With above axiomatized definitions for Klop processes, we first converted them into the following inference rules for transitions which are easier for use:[16]

```
¬(Klop a 0o --u-> E)
Klop a n⁺ --label a-> Klop a n
islimit n ∧ m < n ∧ Klop a m --u-> E ⇒ Klop a n --u-> E
```

Using transfinite induction, we can prove the following properties of the new Klop processes based on ordinals, which is the same with the finite version:

```
Klop_PROP0     ⊢ STABLE (Klop a n)
Klop_PROP1     ⊢ Klop a n --label a-> E ⟺ ∃m. m < n ∧ E = Klop a m
Klop_PROP1_WK  ⊢ Klop a n ==label a=>> E ⟺ ∃m. m < n ∧ E = Klop a m
Klop_PROP2     ⊢ m < n ⇒ ¬(Klop a m ~ Klop a n)
Klop_PROP2_WK  ⊢ m < n ⇒ ¬(Klop a m ≈ Klop a n)
Klop_ONE_ONE   ⊢ ONE_ONE (Klop a)
```

The transfinite induction principles we have used here, is the following two theorems in HOL's `ordinalTheory`:

[16] But these rules alone did not completely capture all the behaviors of Klop processes, because they only talked about the valid transitions and said nothing about invalid transitions

`ord_inductition:`
$\vdash$ $(\forall min.\ (\forall b.\ b < min \Rightarrow P\ b) \Rightarrow P\ min) \Rightarrow \forall \alpha.\ P\ \alpha$
`simple_ord_induction:`
$\vdash$ $P\ 0 \wedge (\forall \alpha.\ P\ \alpha \Rightarrow P\ \alpha^{+}) \wedge$
$(\forall \alpha.$ `islimit` $\alpha \wedge 0 < \alpha \wedge (\forall \beta.\ \beta < \alpha \Rightarrow P\ \beta) \Rightarrow P\ \alpha) \Rightarrow$
$\forall \alpha.\ P\ \alpha$

During the proofs of above properties, many basic results on ordinals were also used, here we omit the proof details.

The next step is to prove the following important result:

Theorem 16. *For any arbitrary set of CCS processes, it's always possible to find a Klop process which is not weakly bisimilar with any processs in the set:*

`INFINITE_KLOP_EXISTS_LEMMA:`
$\vdash$ $\forall a\ A.\ \exists n.\ \forall x.\ x \in A \Rightarrow \neg(x \approx$ `Klop` $a\ n)$

Proof. Our formal proof depends on the following theorem in HOL's `ordinalTheory`:

$\vdash$ $\mathcal{U}$`(:`α `inf)` $\prec$ $\mathcal{U}$`(:`α `ordinal)` `[univ_ord_greater_cardinal]`

which basically says the existence of ordinals larger than the cadinality of any set, which is true in set theory. The HOL type α `inf` means the sum type of `num` and α.

Here we must explain that, we're not living in ZFC or NBG set theory but HOL's own predicated set theory. It's know that, the typed logic implemented in the various HOL systems (including Isabelle/HOL) is not strong enough to define a type for all possible ordinal values (a proper class in a set theory like NBG). Instead, there's a type variable α in ordinals, and to apply above theorem, this type variable must be connected with CCS datatype. Here is the sketch of our formal proof:

We define a mapping f from ordinals to the union of natural numbers and the *power set* of A which actually represents all CCS processes in the graphs of two rooted processes g and h:

$$f(n) = \begin{cases} n & \text{if } n < \omega, \\ \{y\colon y \in B \wedge y \approx Klop_n\} & \text{if } n \geqslant \omega. \end{cases} \tag{6}$$

Suppose the proposition is not true, that is, for each process p in A, there's at least one Klop process k which is weakly bisimlar with p. Then above mapping will never map any ordinal to empty set. And the part for $n < \omega$ is obvious a bijection. And we know the rest part of mapping is one-one.

Now the theorem `univ_ord_greater_cardinal` says there's no injections from ordinals to set A, then there must be at least one non-empty subset of A, and the process in it is weakly bisimilar with two distinct Klop processes. By transitivity of weak equivalence, the two Klop processes must also be weak equivalent, but this violates the property 2 (weak version) of Klop processes. □

A pure set-theory theorem sharing the same proof idea but with all concurrency theorem stuff removed, is the following existence theorem:

Theorem 17. *Assuming an arbitrary set A of type α, and a one-one mapping f from ordinals to type α. There always exists an ordinal n such that $f(n) \notin A$.*

$\vdash$ $\forall$`(`A `:`α `-> bool) (`f `:`α `ordinal -> `α`).`
`ONE_ONE` $f \Rightarrow \exists$`(`n `:`α `ordinal).` $f\ n \notin A$

This result is elegant but unusual, because that "arbitrary set" can simply be the universe of all values of type α, how can there be another value (of the same type) not in it? Our answer is, in such cases the mapping f can't be one-one, and a false assumption will lead to any conclusion in a theorem.[17]

Now we're ready to prove the following full version of "Klop lemma":

[17] On the other side, above theorem seems indicating that, no matter how "complicated" a CCS process is, it's impossible for it to contain all possible equivalence classes of CCS processes as its sub-processes after certain transitions. The "proof" is very similar to the next Klop lemma we present here.

Lemma 6. *(Klop lemma, the full version) For any two CCS processes g and h, there exists another process k which is not weakly equivalent with any sub-process weakly transited from g and h:*

$$\begin{array}{l}\vdash\ \forall g\ h.\\ \quad\exists k.\\ \qquad \texttt{STABLE}\ k\ \wedge\ (\forall g'\ u.\ g\ \texttt{==}u\texttt{=>>}\ g'\ \Rightarrow\ \neg(g'\approx k))\ \wedge\\ \qquad \forall h'\ u.\ h\ \texttt{==}u\texttt{=>>}\ h'\ \Rightarrow\ \neg(h'\approx k)\end{array}\qquad \texttt{[KLOP\_LEMMA]}$$

Proof. We consider the union `nodes` of all nodes (sub-processes) from g and h. If the union is finite, we use previous finite version of this lemma (and the finite version of Klop processes which is well defined in HOL) to get the conclusion. If the union is infinite, we turn to use the full version of Klop process defined (as axiom) on ordinals, and use the previous theorems on ordinals to assert the existence of an ordinal n such that $Klop_n$ is not weakly bisimiar with any node in `nodes`. □

And finally, with *all above lemmas, theorems, definitions, plus one axiomatized definition of infinite Klop process on ordinals*, we have successfully proved the following elegant result without any assumption:

Theorem 18. *(Coarsest congruence contained in $\approx$, the final version) For any processes p and q, $p\ \approx^c\ q$ if and only if $\forall r.\ p + r \approx q + r$.*

9 Conclusions

(to be added)

References

1. Tian, C.: A Formalization of the Process Algebra CCS in HOL4. arXiv.org, `http://arxiv.org/abs/1705.07313v2` (May 2017)
2. Nesi, M.: A formalization of the process algebra CCS in high order logic. Technical report, University of Cambridge, Computer Laboratory (1992)
3. Nesi, M.: Formalising a Value-Passing Calculus in HOL. Formal Aspects of Computing **11**(2) (1999) 160–199
4. Gorrieri, R.: CCS(25, 12) is Turing-complete. submitted for publication
5. Gorrieri, R., Versari, C.: Introduction to Concurrency Theory. Transition Systems and CCS. Springer, Cham (September 2015)
6. Milner, R.: Communication and concurrency. Prentice Hall (1989)
7. van Glabbeek, R.J.: A characterisation of weak bisimulation congruence. Lecture notes in computer science **3838** (2005) 26–39
8. Norrish, M., Huffman, B.: Ordinals in HOL: Transfinite arithmetic up to (and beyond) ω_1. In: International Conference on Interactive Theorem Proving, Springer (2013) 133–146