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. We have added full supports on weak bisimulation equivalence and observation congruence (rooted weak equivalence), with related definitions, theorems and algebraic laws. Some deep lemmas were also formally proved in this project, including Deng Lemma, Hennessy Lemma and several versions of the "Coarsest congruence contained in weak equivalence". For the last theorem, we have proved the full version (without any assumption) based on ordinals.

1 Introduction

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 have formally proved all the strong algebraic laws, including the expansion law. But this is not a complete work for the formalization of CCS, as in most model checking cases, the specification and implementation of the same model has only (rooted) weak equivalence. Thus a further extension to previous work seems meaningful.

This project is still based on old Hol88 proof scripts written by Monica Nesi, but it's not simply a porting of the remain old scripts without new creations. Instead, we have essentially modified many fundamental definitions and have proved many new theorems. And with the changed definitions based on HOL4's rich theory library, previous unprovable theorems now are provable. There're totally 200 theorems and definitions in the project, now about 100 of them were newly defined and proved by the author.

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. [1] As a result, now it's possible to do reasoning on processes with limited number of actions and constants. In academic, the notation $CCS(a,b)$ represents 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. [3]). With the new CCS 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 has no affects, the only exception is the "coarsest congruence contained in $\approx$" (Theorem 4.5 in [4] or Proposition 3 in Chapter 7 of [5]), in which the assumption is not automatically true if the set of labels 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 built-in supports of coinductive relations (`Hol_coreln`) for defining strong and weak bisimulation equivalences. As a result, many intermediate definitions and theorems towards the proof of *Property (*) of strong and weak equivalence* [2] are not needed thus removed from the 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 (especially the right induction theorem) it's impossible to prove the transitivity of observation congruence.

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

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

4. 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 used Deng Lemma to prove the hard part of Hennessy Lemma, therefore minimized the related proof scripts. These results have demonstrated that the author now has ability to convert most informal proofs in CCS into formal proofs.
5. Some important theorems were not proved in the old Hol88 work, notable ones include: 1) the transitivity of observation congruence (`OBS_CONGR_TRANS`) and 2) the coarsest congruence containing in weak equivalence (`PROP3`). In this project, we have finished these proofs with new related lemmas stated 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 [6] published by J. R. van Glabbeek in 2005. Then we use HOL's ordinal theory and an axiomatized support of infinite sums of CCS to prove the full version for general CCS processes.
6. A rather complete theory of congruence on CCS has been newly built in this project. We have used this theory to explain the meaning of " coarsest congruence containing in weak equivalence" theorem.

With these new additions, now the whole project has established the comprehensive theory for pure CCS with all major results included. The project can either be used as the theoretical basis for discovering new results about CCS, or further developed into a model checking software running in theorem prover.

This project is now part of official HOL4 repository[3], under the directory `examples/CCS`.

2 Background

Our work in this and previous project were based on an old CCS formalization [7] built in Hol88 theorem prover (ancestry of HOL4), by Monica Nesi during 1992-1995. As noted already in Background section of the report of our previous project, 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 beginning 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 [4] under the supervision of Prof. Roberto Gorrieri in University of Bologna. And instead of creating everything from scratch, the author has 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.

[3] `https://github.com/HOL-Theorem-Prover/HOL`

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

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 important 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 as "`(`α`,` β`) CCS`". Whenever both type variables were instantiated as `string`, the resulting type "`(string, string) CCS`" is equivalent with the `CCS` datatype in previous project. Within the new settings, to represent CCS subcalculus like $CCS(25, 12)$, custom datatypes with limited number of instances can be defined by users[5]. However we didn't go further in this direction.

The type of transition labels were extended 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 value, 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 omited from the assumptions of the theorem, because it's automatically satisfied.

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 [4]) 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` with the helper definition `EPS0` removed[6]:

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' ($E \xrightarrow{\tau} E'$):*[7]

$\vdash$ `EPS = (`$\lambda E\ E'$`.` E `--`τ`->` E'`)`*

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

Sometimes it's necessary to consider different transition cases when $p \overset{\epsilon}{\Rightarrow} q$ 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)*

[5] 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.

[6] After the author has learnt to use λ-expressions to express relations

[7] 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.

$\vdash x \overset{\epsilon}{\Rightarrow} y \iff x = y \lor \exists u.\ x \text{ --}\tau\text{-> } u \land u \overset{\epsilon}{\Rightarrow} y$ [EPS_cases1]
$\vdash x \overset{\epsilon}{\Rightarrow} y \iff x = y \lor \exists u.\ x \overset{\epsilon}{\Rightarrow} u \land u \text{ --}\tau\text{-> } y$ [EPS_cases2]
$\vdash E \overset{\epsilon}{\Rightarrow} E' \iff E \text{ --}\tau\text{-> } E' \lor E = E' \lor \exists E_1.\ E \overset{\epsilon}{\Rightarrow} E_1 \land E_1 \overset{\epsilon}{\Rightarrow} E'$ [EPS_cases]

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

$\vdash (\forall x.\ P\ x\ x) \land (\forall x\ y\ z.\ x \text{ --}\tau\text{-> } y \land P\ y\ z \Rightarrow P\ x\ z) \Rightarrow$
$\quad \forall x\ y.\ x \overset{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_ind]
$\vdash (\forall x.\ P\ x\ x) \land (\forall x\ y\ z.\ x \text{ --}\tau\text{-> } y \land y \overset{\epsilon}{\Rightarrow} z \land P\ y\ z \Rightarrow P\ x\ z) \Rightarrow$
$\quad \forall x\ y.\ x \overset{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_strongind]
$\vdash (\forall x.\ P\ x\ x) \land (\forall x\ y\ z.\ P\ x\ y \land y \text{ --}\tau\text{-> } z \Rightarrow P\ x\ z) \Rightarrow$
$\quad \forall x\ y.\ x \overset{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_ind_right]
$\vdash (\forall x.\ P\ x\ x) \land (\forall x\ y\ z.\ P\ x\ y \land x \overset{\epsilon}{\Rightarrow} y \land y \text{ --}\tau\text{-> } z \Rightarrow P\ x\ z) \Rightarrow$
$\quad \forall x\ y.\ x \overset{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_strongind_right]
$\vdash (\forall E\ E'.\ E \text{ --}\tau\text{-> } E' \Rightarrow P\ E\ E') \land (\forall E.\ P\ E\ E) \land$
$\quad (\forall E\ E_1\ E'.\ P\ E\ E_1 \land P\ E_1\ E' \Rightarrow P\ E\ E') \Rightarrow$
$\quad \forall x\ y.\ x \overset{\epsilon}{\Rightarrow} y \Rightarrow P\ x\ y$ [EPS_INDUCT]

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: $E \overset{a}{\longrightarrow} E'$ if and only if there exists two processes E_1 and E_2 such that $E \overset{\epsilon}{\Longrightarrow} E_1 \overset{a}{\longrightarrow} E_2 \overset{\epsilon}{\Longrightarrow} E'$:*

$\vdash E \text{ ==}u\text{=>> } E' \iff \exists E_1\ E_2.\ E \overset{\epsilon}{\Rightarrow} E_1 \land E_1 \text{ --}u\text{-> } E_2 \land E_2 \overset{\epsilon}{\Rightarrow} E'$ [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:*

   $\vdash E \text{ --}u\text{-> } E' \Rightarrow E \text{ ==}u\text{=>> } E'$ [TRANS_IMP_WEAK_TRANS]

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

   $\vdash E \text{ ==}\tau\text{=>> } E' \Rightarrow E \overset{\epsilon}{\Rightarrow} E'$ [WEAK_TRANS_TAU]

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

   $\vdash E \text{ --}\tau\text{-> } E' \Rightarrow E \overset{\epsilon}{\Rightarrow} E'$ [TRANS_TAU_IMP_EPS]

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

   $\vdash E \text{ ==}\tau\text{=>> } E' \Rightarrow \exists E_1.\ E \text{ --}\tau\text{-> } E_1 \land E_1 \overset{\epsilon}{\Rightarrow} E'$ [WEAK_TRANS_TAU_IMP_TRANS_TAU]

5. EPS *implies τ-prefixed* EPS*:*

   $\vdash E \overset{\epsilon}{\Rightarrow} E' \Rightarrow \tau..E \overset{\epsilon}{\Rightarrow} E'$ [TAU_PREFIX_EPS]

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

   $\vdash E \text{ ==}u\text{=>> } E' \Rightarrow \tau..E \text{ ==}u\text{=>> } E'$ [TAU_PREFIX_WEAK_TRANS]

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

   $\vdash E \overset{\epsilon}{\Rightarrow} E_1 \land E_1 \text{ ==}u\text{=>> } E_2 \land E_2 \overset{\epsilon}{\Rightarrow} E' \Rightarrow E \text{ ==}u\text{=>> } E'$ [EPS_AND_WEAK]

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

   $\vdash E \text{ --}\tau\text{-> } E_1 \land E_1 \text{ ==}u\text{=>> } E' \Rightarrow E \text{ ==}u\text{=>> } E'$ [TRANS_TAU_AND_WEAK]

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

   $\vdash E \text{ --}u\text{-> } E_1 \land E_1 \overset{\epsilon}{\Rightarrow} E' \Rightarrow E \text{ ==}u\text{=>> } E'$ [TRANS_AND_EPS]

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

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

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

$$\begin{array}{l} \vdash E \texttt{ ==}u\texttt{=>> } E_1 \Rightarrow \\ \quad (\exists E'.\ E \texttt{ --}\tau\texttt{-> } E' \land E' \texttt{ ==}u\texttt{=>> } E_1) \lor \\ \quad \exists E'.\ E \texttt{ --}u\texttt{-> } E' \land E' \overset{\epsilon}{\Rightarrow} E_1 \end{array} \qquad \texttt{[WEAK\_TRANS\_cases1]}$$

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 equivalence

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.

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 [4], except for the use of EPS in case of τ-transitions:

Definition 3. *(Weak bisimulation)*

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

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:

$\vdash$ WEAK_BISIM $Wbsm$ $\Rightarrow$ WEAK_BISIM $(\lambda x\ y.\ Wbsm\ y\ x)$ [IDENTITY_WEAK_BISIM]

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

$\vdash$ WEAK_BISIM $Wbsm_1$ $\land$ WEAK_BISIM $Wbsm_2$ $\Rightarrow$
WEAK_BISIM $(\lambda x\ z.\ \exists y.\ Wbsm_1\ x\ y \land Wbsm_2\ y\ z)$ [COMP_WEAK_BISIM]

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

$\vdash$ WEAK_BISIM $Wbsm_1$ $\land$ WEAK_BISIM $Wbsm_2$ $\Rightarrow$
WEAK_BISIM $(\lambda x\ y.\ Wbsm_1\ x\ y \lor Wbsm_2\ x\ y)$ [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':*

$\vdash$ $E \approx E' \iff \exists Wbsm.\ Wbsm\ E\ E' \land$ 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 [5]:

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

$$
\begin{aligned}
&\vdash a_0 \approx a_1 \iff \\
&\quad (\forall l. \\
&\qquad (\forall E_1. \\
&\qquad\quad a_0 \texttt{ --label } l\texttt{-> } E_1 \Rightarrow \\
&\qquad\quad \exists E_2.\ a_1 \texttt{ ==label } l\texttt{=>> } E_2 \land E_1 \approx E_2) \land \\
&\qquad \forall E_2. \\
&\qquad\quad a_1 \texttt{ --label } l\texttt{-> } E_2 \Rightarrow \\
&\qquad\quad \exists E_1.\ a_0 \texttt{ ==label } l\texttt{=>> } E_1 \land E_1 \approx E_2) \land \\
&\quad (\forall E_1.\ a_0 \texttt{ --}\tau\texttt{-> } E_1 \Rightarrow \exists E_2.\ a_1 \overset{\epsilon}{\Rightarrow} E_2 \land E_1 \approx E_2) \land \\
&\quad \forall E_2.\ a_1 \texttt{ --}\tau\texttt{-> } E_2 \Rightarrow \exists E_1.\ a_0 \overset{\epsilon}{\Rightarrow} E_1 \land E_1 \approx E_2 \qquad \texttt{[OBS\_PROPERTY\_STAR]}
\end{aligned}
$$

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 (*).[8]

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)*

$\vdash$ `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 substitutability 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 substitutability theorems we must added extra assumptions on the processes involved, i.e. the stability 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:*

$\vdash \texttt{STABLE}\ E \iff \forall u\ E'.\ E\ \texttt{--}u\texttt{->}\ E' \Rightarrow u \neq \tau$

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:*

   $\vdash E_1 \approx E_1' \wedge \texttt{STABLE}\ E_1 \wedge \texttt{STABLE}\ E_1' \wedge E_2 \approx E_2' \wedge \texttt{STABLE}\ E_2 \wedge$
   $\texttt{STABLE}\ E_2' \Rightarrow$
   $E_1 + E_2 \approx E_1' + E_2'$ [WEAK_EQUIV_PRESD_BY_SUM]

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

   $\vdash E \approx E' \wedge \texttt{STABLE}\ E \wedge \texttt{STABLE}\ E' \Rightarrow \forall E''.\ E + E'' \approx E' + E''$

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

   $\vdash E \approx E' \wedge \texttt{STABLE}\ E \wedge \texttt{STABLE}\ E' \Rightarrow \forall E''.\ E'' + E \approx E'' + E'$

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]

[8] 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.

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

$\vdash E \approx E' \Rightarrow \forall rf.\ \texttt{relab}\ E\ rf \approx \texttt{relab}\ E'\ rf$ [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

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:*

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

$\vdash E \approx^c E' \Rightarrow E \approx E'$ [OBS_CONGR_IMP_WEAK_EQUIV]

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

$\vdash E \approx E' \land \texttt{STABLE}\ E \land \texttt{STABLE}\ E' \Rightarrow E \approx^c E'$ [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)*

$\vdash E \approx^c E$ [OBS_CONGR_REFL]
$\vdash E \approx^c E' \Rightarrow E' \approx^c E$ [OBS_CONGR_SYM]

But the transitivity is hard to prove.[9] 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:*

$\vdash E \approx^c E' \Rightarrow \forall E_1.\ E \stackrel{\epsilon}{\Rightarrow} E_1 \Rightarrow \exists E_2.\ E' \stackrel{\epsilon}{\Rightarrow} E_2 \land E_1 \approx E_2$ [OBS_CONGR_EPS]

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

Proof. By (right) induction[10] 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:

$$\begin{tikzcd} E \arrow[r, dotted, no head, "\approx^c"] \arrow[d, no head, "="] & E' \arrow[d, no head, "="] \\ E \arrow[r, dotted, no head, "\approx"] & E' \end{tikzcd}$$

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 $E \overset{\epsilon}{\Rightarrow} E_1$, $E' \overset{\epsilon}{\Rightarrow} E_2$ and $E_1 \approx E_2$. Now by definition of weak equivalence, if E_1 -τ-> E_1' then there exists $E2'$ such that $E_2 \overset{\epsilon}{\Rightarrow} E_2'$ and $E_1' \approx E_2'$. Then by transitivity of EPS, we have $E' \overset{\epsilon}{\Rightarrow} E_2 \wedge E_2 \overset{\epsilon}{\Rightarrow} E_2' \Rightarrow E' \overset{\epsilon}{\Rightarrow} E_2'$, thus E_2' is a valid response required by observation congruence:

$$\begin{tikzcd} E \arrow[r, dotted, no head, "\approx^c"] \arrow[d, Rightarrow, "\epsilon"] & E' \arrow[d, Rightarrow, "\epsilon"] \arrow[dd, bend left, "\epsilon"] \\ \forall E_1 \arrow[r, dotted, no head, "\approx"] \arrow[d, "\tau"] & \forall E_2 \arrow[d, Rightarrow, "\epsilon"] \\ \forall E_1' \arrow[r, dotted, no head, "\approx"] & \exists E_2' \end{tikzcd}$$

□

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:*

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

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:

$$\begin{tikzcd} E \arrow[r, dotted, no head, "\approx^c"] \arrow[d, "\tau"] \arrow[dd, bend right, "\tau"'] & E' \arrow[d, Rightarrow, "\tau"] \arrow[dd, bend left, "\tau"] \\ \exists E_1' \arrow[r, dotted, no head, "\approx"] \arrow[d, Rightarrow, "\epsilon"] & \exists E_2 \arrow[d, Rightarrow, "\epsilon"] \\ \forall E_1 \arrow[r, dotted, no head, "\approx"] & \exists E_2' \end{tikzcd}$$

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:

$$\begin{tikzcd} E \arrow[r, dotted, no head, "\approx^c"] \arrow[d, Rightarrow, "\epsilon"] \arrow[ddd, bend right, "\forall L"'] & E' \arrow[d, Rightarrow, "\epsilon"] \arrow[ddd, bend left, "L"] \\ \exists E_1' \arrow[r, dotted, no head, "\approx"] \arrow[d, "L"] & \exists E_2' \arrow[d, Rightarrow, "L"] \\ \exists E_2 \arrow[r, dotted, no head, "\approx"] \arrow[d, Rightarrow, "\epsilon"] & \exists E_2'' \arrow[d, Rightarrow, "\epsilon"] \\ \forall E_1 \arrow[r, dotted, no head, "\approx"] & \exists E2''' \end{tikzcd}$$

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

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. $\square$

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

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

Now we prove the transitivity of observation congruence:

Theorem 5. *(Transitivity of Observation Congruence)*

$\vdash\ E\ \approx^c\ E'\ \wedge\ E'\ \approx^c\ E''\ \Rightarrow\ E\ \approx^c\ E''$ [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 $E\ \texttt{-}u\texttt{->}\ E_1$, by definition of observation congruence, there exists E_2 such that $E'\ \texttt{==}u\texttt{=}\!\gg\ E_2$ with $E_1\ \approx\ E_2$. By above Lemma 2, there exists another E_3 such that $E''\ \texttt{==}u\texttt{=}\!\gg\ E_3$ 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 satisfies the definition of observation congruence. This proves the first part. The other part is completely symmetric.

$$
\begin{tikzcd}
\forall E1 \arrow[r, dotted, no head, "\approx"] \arrow[rr, bend left, "\approx(goal)"] & \exists E_2 \arrow[r, dotted, no head, "\approx"] & \exists E_3 \\
\forall E \arrow[u, "\forall u"] \arrow[r, dotted, no head, "\approx^c"] & E' \arrow[u, Rightarrow, "u"] \arrow[r, dotted, no head, "\approx^c"] & E'' \arrow[u, Rightarrow, "u"]
\end{tikzcd}
$$

$\square$

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:*

$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ \forall\, u\texttt{.}\ \ u\texttt{..}E\ \approx^c\ u\texttt{..}E'$ [OBS_CONGR_SUBST_PREFIX]

2. *Observation congruence is substitutive under binary summation:*

$\vdash\ p\ \approx^c\ q\ \wedge\ r\ \approx^c\ s\ \Rightarrow\ p\ \texttt{+}\ r\ \approx^c\ q\ \texttt{+}\ s$ [OBS_CONGR_PRESD_BY_SUM]

3. *Observation congruence is preserved by parallel composition:*

$\vdash\ E_1\ \approx^c\ E_1'\ \wedge\ E_2\ \approx^c\ E_2'\ \Rightarrow\ E_1\ \texttt{||}\ E_2\ \approx^c\ E_1'\ \texttt{||}\ E_2'$ [OBS_CONGR_PRESD_BY_PAR]

4. *Observation congruence is substitutive under the restriction operator:*

$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ \forall\, L\texttt{.}\ \ \nu\ \ L\ \ E\ \approx^c\ \nu\ \ L\ \ E'$ [OBS_CONGR_SUBST_RESTR]

5. *Observation congruence is substitutive under the relabeling operator:*

$\vdash\ E\ \approx^c\ E'\ \Rightarrow\ \forall\, \mathit{rf}\texttt{.}\ \ \texttt{relab}\ \ E\ \ \mathit{rf}\ \approx^c\ \texttt{relab}\ \ E'\ \ \mathit{rf}$ [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)*

$\vdash\ E\ \sim\ E'\ \Rightarrow\ E\ \approx^c\ E'$ [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)*

$\vdash\ u\texttt{..}\tau\texttt{..}E\ \approx^c\ u\texttt{..}E$ [TAU1]
$\vdash\ E\ \texttt{+}\ \tau\texttt{..}E\ \approx^c\ \tau\texttt{..}E$ [TAU2]
$\vdash\ u\texttt{..(}E\ \texttt{+}\ \tau\texttt{..}E'\texttt{)}\ \texttt{+}\ u\texttt{..}E'\ \approx^c\ u\texttt{..(}E\ \texttt{+}\ \tau\texttt{..}E'\texttt{)}$ [TAU3]
$\vdash\ E\ \texttt{+}\ \tau\texttt{..(}E'\ \texttt{+}\ E\texttt{)}\ \approx^c\ \tau\texttt{..(}E'\ \texttt{+}\ E\texttt{)}$ [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[11]):

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. $p \approx^c q$.

$\vdash p \approx q \Rightarrow$
$(\exists p'.\ p$ `--`τ`->` $p' \wedge p' \approx q) \vee (\exists q'.\ q$ `--`τ`->` $q' \wedge p \approx q') \vee$
$p \approx^c q$ [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 $q \overset{\epsilon}{\Rightarrow} E_2$ and $E_1 \approx E_2$. But by assumption we know $q \neq E2$, thus $q \overset{\epsilon}{\Rightarrow} E_2$ contains at least one τ-transition, thus is actually q `==`τ`=»` E_2, which is required by the definition of observation congruence for $p \approx q$.

$$\begin{tikzcd} p \arrow[r, dotted, no head, "\approx^c"] \arrow[d, "\tau"] & q \arrow[d, Rightarrow, "\epsilon(\tau)"] \arrow[dl, dotted, no head, "\not\approx"'] \\ \forall E_1 \arrow[r, dotted, no head, "\approx"'] & \exists E_2 \end{tikzcd}$$

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* $\tau..p \approx^c q$*):*

$\vdash p \approx q \iff p \approx^c q \vee p \approx^c \tau..q \vee \tau..p \approx^c q$ [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 [4], 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 $p \approx^c q$.

[11] 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 mathematics theories, the author has got the ability to express it in HOL theorem prover.

8 The theory of congruence

The highlight of this project is the formal proofs for various versions of the "coarsest congruence contained in weak equivalence",

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

But at first glance, the name of above theorem doesn't make much sense. To see the nature of above theorem more clearly, here we represent a rather complete theory about the congruence of CCS. It's based on contents from [6].

To formalize the concept of congruence, we need to define "semantic context" first. There're multiple solutions, here we have chosen a simple solution based on λ-calculus:

Definition 7. *(Semantic context of CCS) The semantic context (or one-hole context) of CCS is a function $C[\cdot]$ of type "(α, β)* `CCS -> (`α, β`) CCS`*" recursively defined by following rules:*

$$
\begin{array}{l}
\texttt{CONTEXT}\ (\lambda x.\ x) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ a..c\ t) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ c\ t\ +\ x) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ x\ +\ c\ t) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ c\ t\ ||\ x) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ x\ ||\ c\ t) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ \nu\ L\ (c\ t)) \\
\texttt{CONTEXT}\ c \Rightarrow \texttt{CONTEXT}\ (\lambda t.\ \texttt{relab}\ (c\ t)\ rf)
\end{array}
$$

By repeatedly applying above rules, one can imagine that, a "hold" in any CCS term at any depth, can become a λ-function, and by calling the function with another CCS term, the hold is filled by that term.

The notable property of one-hole context is that, the functional combination of two contexts is still a context:

Proposition 11. *(The combination of one-hole contexts) If both c_1 and c_2 are one-hole contexts, then $c_1 \circ c_2$[12] is still a one-hole context:*

$\vdash$ `CONTEXT` c_1 $\land$ `CONTEXT` c_2 $\Rightarrow$ `CONTEXT` $(c_1 \circ c_2)$

Proof. By induction on the first context c_1. □

Now we're ready to define the concept of congruence (for CCS):

Definition 8. *(Congruence of CCS) An equivalence relation $\approx$[13] on a specific space of CCS processes is a* congruence *iff for every n-ary operator f, one has $g_1 \approx h_1 \land \cdots g_n \approx h_n \Rightarrow f(g_1, \ldots, g_n) \approx f(h_1, \ldots, h_n$. This is the case iff for every semantic context $C[\cdot]$ on has $g \approx h \Rightarrow C[g] \approx C[h]$:*

$$
\begin{array}{l}
\vdash \texttt{congruence}\ R \iff \\
\quad \forall x\ y\ ctx.\ \texttt{CONTEXT}\ ctx \Rightarrow R\ x\ y \Rightarrow R\ (ctx\ x)\ (ctx\ y)
\end{array}
$$

We can easily prove that, strong equivalence and observation congruence is indeed a congruence following above definition, using the substitutability and preserving properties of these relations:

Theorem 10. $\vdash$ `congruence STRONG_EQUIV`
$\vdash$ `congruence OBS_CONGR`

[12] $(c_1 \circ c_2)t := c_1(c_2 t)$.
[13] The symbol $\approx$ here shouldn't be understood as weak equivalence.

For relations which is not congruence, it's possible to "convert" them into congruence:

Definition 9. *(Constructing congruences from equivalence relation) Given an equivalence relation* $\sim$[14]*, define* $\sim^c$ *by:*

$\vdash R^c$ `=` $(\lambda\, g\ h.\ \forall\, c.$ `CONTEXT` $c \Rightarrow R\ (c\ g)\ (c\ h))$

This new operator on relations has the following three properties:

Proposition 12. *For all* R*,* R^c *is a congruence:*

$\vdash$ `congruence` R^c

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 13. *For all* R*,* R^c *is finer than* R*:*

$\vdash R^c \subseteq_r R$

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

Proposition 14. *For all* R*,* R^c *is the coarsest congruence finer than* R*, that is, for any other congruence finer than* R*, it's finer than* R^c*:*

$\vdash$ `congruence` $R' \wedge R' \subseteq_r R \Rightarrow R' \subseteq_r R^c$

Proof. If $\approx$ is any congruence finer than $\sim$, then

$$g \approx h \Rightarrow \forall C[\cdot].\ (C[g] \approx C[h]) \Rightarrow \forall C[\cdot].\ (C[g] \sim C[h]) \Rightarrow g \sim^c h. \tag{1}$$

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

As we know weak equivalence is not congruence, and one way to "fix" it, is to use observation congruence which is based on weak equivalence but have special treatments on the first transitions. The other way is to build a congruence from existing weak equivalence relation, using above approach based on one-hole contexts. Such a congruence has a new name:

Definition 10. *(Weak bisimulation congruence) The coarasest congruence that is finer than weak bisimulation equivalence is called* weak bisimulation congruence *(notation:* $\sim^c_w$*):*

$\vdash$ `WEAK_CONGR = (=~)`^{c}
or
$\vdash$ `WEAK_CONGR` `=` $(\lambda\, g\ h.\ \forall\, c.$ `CONTEXT` $c \Rightarrow c\ g \approx c\ h)$

So far, the weak bisimulation congruence $\sim^c_w$ 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:

$\vdash E \approx^c E' \iff$
  $\forall\, u.$
    $(\forall\, E_1.\ E$ `--`u`->` $E_1 \Rightarrow \exists\, E_2.\ E'$ `==`u`=>>` $E_2 \wedge E_1 \approx E_2) \wedge$
    $\forall\, E_2.\ E'$ `--`u`->` $E_2 \Rightarrow \exists\, E_1.\ E$ `==`u`=>>` $E_1 \wedge E_1 \approx E_2$

But since obvervation congruence is congruence, it must be finer than weak bisimulation congruence:

Lemma 4. *(Obvervation congruence is finer than weak bisimulation congruence)*

$\vdash p \approx^c q \Rightarrow$ `WEAK_CONGR` $p\ q$

[14] The Symbol $\sim$ here shouldn't be understood as strong equivalence.

On the other side, by consider the trivial context and sum contexts in the definition of weak bisimulation congruence, we can easily prove the following result:

Lemma 5. $\vdash$ `WEAK_CONGR` $p\ \ q\ \Rightarrow$ `SUM_EQUIV` $p\ \ q$

Noticed that, in above theorem, the sum operator can be replaced by any other operator in CCS, but we know sum is special because it's the only operator in which the weak equivalence is not preserved after substitutions.

From above two lemmas, we can easily see that, weak equivalence is between the observation congruence and an unnamed relation $\{(p, q) \colon \forall r.p + r \approx q + r\}$ (we can temporarily call it "sum equivalence", because we don't if it's a congruence, or even if it's contained in weak equivalence). If we could further prove that "sum equivalence" is finer than observation congruence:

Proposition 15. $\vdash\ (\forall\, r.\ \ p\ +\ r\ \approx\ q\ +\ r)\ \Rightarrow\ p\ \approx^c\ q$

then all three congruences (observation congruence, weak equivalence and the "sum equivalence" must all coincide, as illustrated in the following figure:

$$
\begin{tikzcd}
\text{Weak equivalence}(\approx) & & \text{Sum equivalence} \arrow[ddl, bend left, "\subseteq?"] \\
& \text{Weak bisimulation congruence}(\sim^c_w) \arrow[ul, "\subseteq"] \arrow[ur, "\subseteq"'] & \\
& \text{Observation congruence}(\approx^c) \arrow[u, "\subseteq"] &
\end{tikzcd}
$$

This is why the proposition at the begining of this section is called "coarsest congruence contained in weak equivalence", it's actually trying to prove the "sum equivalence" is finer than "observation congruence" therefore makes "weak bisimulation congruence" ($\sim^c_w$) coincide with "observation congruence" ($\approx^c$).

9 Coarsest congruence contained in weak equivalence

The hightlight of this project is the formal proofs of various versions of the so-called "coarsest congruence contained in $\approx$" theorem:

Proposition 16. $\vdash\ p\ \approx^c\ q\ \iff\ \forall\, r.\ \ p\ +\ r\ \approx\ q\ +\ r$

It's surprising hard to prove this result, when there's no assumptions on the processes. We consider the following three cases with increasing difficulities:

1. with classical cardinality assumptions;
2. for finite state CCS;
3. general case.

The easy part (left $\Longrightarrow$ right) is already proven in previous section by combining `OBS_CONGR_-IMP_WEAK_CONGR` and `WEAK_CONGR_IMP_SUM_EQUIV`, or it can be proved directly using `OBS_-CONGR_IMP_WEAK_EQUIV` and `OBS_CONGR_SUBST_SUM_R`:

Theorem 11. *(The easy part "Coarsest congruence contained in $\approx$")*

$\vdash\ p\ \approx^c\ q\ \Rightarrow\ \forall\, r.\ \ p\ +\ r\ \approx\ q\ +\ r$ [COARSEST_CONGR_LR]

Thus we only focus on the hard part (right $\Longrightarrow$ left) in the rest of this section.

9.1 With classicial cardinality assumptions

A classic restriction is to assume cardinality limitations on the two processes, so that didn't use up all possible labels. Sometimes this assumption is automatically satisfied, for example: the CCS is finitrary and the set of all actions is infinite. But in our setting, the CCS datatype contains twi type variables, and if the set of all possible labels has only finite cardinalities, this assumtion may not be satisfied.

In [5] (Proposition 3 in Chapter 7, p. 153), Robin Milner simply called this theorem the "Proposition 3":

Proposition 17. *(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 [4] (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 12. *(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. Instead, it can be weakened to the following stronger version, which assumes the following properties instead:

Definition 11. *(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:*

$\vdash$ `free_action` $p \iff \exists\, a.\ \forall\, p'.\ \neg($$p$ `==label` a`=>>` $p'$$)$

Theorem 13. *(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 $r \in \mathscr{P}$:*

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

This new assumption is weaker 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.

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 $p + a.0 \overset{u}{\Longrightarrow} E_1$, 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):

$$\begin{tikzcd} p + a.0 \arrow[r, dotted, no head, "\approx"] \arrow[d, "u=\tau"] & q + a.0 \arrow[d, Rightarrow, "\epsilon"] \\ E_1 \arrow[r, dotted, no head, "\approx"] & E_2 \end{tikzcd}$$

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 $E_1 \overset{a}{\Longrightarrow} E_1'$, but this means $p \overset{a}{\Longrightarrow} E_1'$, which is impossible by free-action assumption on p:

$$
\begin{tikzcd}
p \arrow[dr, "\tau"] \arrow[ddr, "a"'] & p + a.0 \arrow[d, "\tau"] \arrow[r, dotted, no head, "\approx"] & q + a.0 = E_2 \arrow[d, "a"] \\
 & E_1 \arrow[d, Rightarrow, "a"] \arrow[ur, dotted, no head, "\approx"] & 0 \\
 & E_1' \arrow[ur, dotted, no head, "\approx"] &
\end{tikzcd}
$$

Once we have $q + a.0 \overset{\tau}{\Longrightarrow} E2$, 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 $q + a.0 \overset{L}{\Longrightarrow} E_2$. We can further conclude that $q \overset{L}{\Longrightarrow} E_2$ 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. $\square$

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

Theorem 14. *(Coarsest congruence contained in $\approx$)*

```
⊢ free_action p ∧ free_action q ⇒ (p ≈ᶜ q  ⟺  ∀r. p + r ≈ q + r)
```

9.2 Without cardinality assumptions

In 2005, Rob J. van Glabbeek published a paper [6] 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. This proof is not known to Robin Milner in [5]. [15]

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

Theorem 15. *(Coarsest congruence contained in $\approx$, new assumptions) For any two CCS processes p and q, 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 $p \approx^c q$ if $p + r \approx q + r$ for all $r \in \mathscr{P}$.*

```
⊢ (∃k.
     STABLE k ∧ (∀p′ u. p ==u=>> p′ ⇒ ¬(p′ ≈ k)) ∧
     ∀q′ u. q ==u=>> q′ ⇒ ¬(q′ ≈ k)) ⇒
  (∀r. p + r ≈ q + r) ⇒
  p ≈ᶜ q
```

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 $p + a.k \approx q + a.k$ as our working basis. For all transitions from p, say $p \xrightarrow{u} E_1$, we're going to prove that, there must be a corresponding weak transition such that $q \overset{u}{\Longrightarrow} E_2$, and $E_1 \approx E_2$ (thus $p \approx^c q$. There're three cases to consider:

1. τ-transitions: $p \xrightarrow{\tau} E_1$. By SOS rule (Sum_1), we have $p + a.k \xrightarrow{\tau} E_1$, now by $p + a.k \approx q + a.k$ and the property (*) of weak equivalence, there exists an E_2 such that $q + a.k \overset{\epsilon}{\Longrightarrow} E_2$. 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 $q + a.k = E_2$, and since $E_1 \approx E_2$, for transition $q + a.k \xrightarrow{a} k$,

[15] 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.

E_1 must make a response by $E_1 \overset{a}{\Longrightarrow} E_1'$, and as the result we have $p \overset{a}{\Longrightarrow} E_1'$ and $E_1' \approx k$, which is impossible by the special choice of k:

$$\begin{tikzcd}
p \arrow[dr, "\tau"] \arrow[ddr, Rightarrow, "a"'] & p + a.k \arrow[r, dotted, no head, "\approx"] \arrow[d, "\tau"] & q + a.k = E_2 \arrow[d, "a"] \arrow[dl, dotted, no head, "\approx"'] \\
& E_1 \arrow[d, Rightarrow, "a"] & k \arrow[dl, dotted, no head, "\not\approx"'] \\
& E_1' &
\end{tikzcd}$$

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

$$\begin{tikzcd}
p \arrow[dr, "a"'] & p + a.k \arrow[r, dotted, no head, "\approx"] \arrow[d, "a"] & q + a.k \arrow[d, Rightarrow, "a"] \\
& \forall E_1 \arrow[r, dotted, no head, "\approx"] & \exists E_2
\end{tikzcd}$$

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 q (otherwise the first step would be an a-transition from $a.k$). And in this case we can easily conclude $q \overset{a}{\Longrightarrow} E_2$ without using the property of k:

$$\begin{tikzcd}
p \arrow[rrr, bend left, "\approx^c"] \arrow[dr, "a"'] & p + a.k \arrow[r, dotted, no head, "\approx"] \arrow[d, "a"] & q + a.k \arrow[d, "\tau"] & q \arrow[dl, "\tau"'] \arrow[ddl, Rightarrow, "a"] \\
& \forall E_1 \arrow[dr, dotted, no head, "\approx"] & \exists E' \arrow[d, Rightarrow, "a"] & \\
& & \exists E_2 &
\end{tikzcd}$$

(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:

$$\begin{tikzcd}
p \arrow[dr, "a"'] & p + a.k \arrow[r, dotted, no head, "\approx"] \arrow[d, "a"] & q + a.k \arrow[d, "a"] \\
& \forall E_1 \arrow[r, dotted, no head, "\not\approx"] & E2 = k
\end{tikzcd}$$

3. For other L-transitions coming from p, where $L \neq a$ and $L \neq \tau$. As a response to $p+a.k \xrightarrow{L} E_1$, we have $q + a.k \overset{L}{\Longrightarrow} E_2$ and $E_1 \approx E_2$. It's obvious that $q \overset{L}{\Longrightarrow} E_2$ 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:

$$\begin{tikzcd}
p \arrow[rrr, bend left, "\approx^c"] \arrow[dr, "\forall L"'] & p + a.k \arrow[r, dotted, no head, "\approx"] \arrow[d, "L"] & q + a.k \arrow[d, Rightarrow, "L"] & q \arrow[dl, Rightarrow, "L"] \\
& \forall E_1 \arrow[r, dotted, no head, "\approx"] & \exists E2 &
\end{tikzcd}$$

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

Now it remains to prove the existence of the special process mentioned in the assumption of above theorem.

9.3 Arbitrary many non-bisimilar processes

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

The assumption in previous Theorem 15 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[16] Thus there's no essential differences to consider all states in the process group instead.

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

Definition 12. *(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[17] without intruducing new axioms. Therefore we have followed a two-step approach in this project: first we consider only the finite-state CCS (no need for axioms), then we turn to the general case.

9.4 Finite-state CCS

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 13. *(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 18. *(Properties of Klop functions and processes)*

1. *All Klop processes are stable:*

   ⊢ `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:*

[16] 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.

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

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

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

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

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']

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 map them into a set containing all these Klop processes, and the set is countable infinite. On the other side, the number of all states coming 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 6. *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:*

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

Proof. We built an explicit mapping f from A to B[18], 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 $x \in A$, 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 violates 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 x (in A) which is equivalent with k. □

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

Definition 14. *(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'`)`*

[18] 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.

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 19. *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 7. *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:*

```
⊢ ∀p q.
    FINITE_STATE p ∧ FINITE_STATE q ⇒
    ∃k.
      STABLE k ∧ (∀p′ u. p ==u=>> p′ ⇒ ¬(p′ ≈ k)) ∧
      ∀q′ u. q ==u=>> q′ ⇒ ¬(q′ ≈ k)                [KLOP_LEMMA_FINITE]
```

Combining above lemma, Theorem 15 and Theorem 11, we can easily prove the following theorem for finite-state CCS:

Theorem 16. *(Coarsest congruence contained in ≈ for finite-state CCS)*

```
⊢ FINITE_STATE p ∧ FINITE_STATE q ⇒
  (p ≈^c q ⟺ ∀r. p + r ≈ q + r)
```

9.5 General case

Now we turn to the general case.[19]. 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, due to limitations in higher order logic, the CCS datatype has no way to express infinite sum of CCS processes, e.g. an constructor `(summ :((`α`, `β`) CCS -> bool) -> (`α`, `β`) CCS)`.[20] As a result, we had no choice but to introduce a new "axiom" for reasoning about infinite sums of CCS:

Proposition 20. *(Infinite sum axiom for CCS)*

[19] This part in the project is not commited into HOL official repository, because of the possibly wrong use of `new_axiom`, full project code can be found at `https://github.com/binghe/informatica-public/tree/master/CCS2`

[20] Michael Norrish, HOL maintainer, explains the reason: "You can't define a type that recurses under the set 'constructor' (your summ constructor has (CCS set) as an argument). Ignoring the num set argument, you would then have an injective function (the summ constructor itself) from sets of CCS values into single CCS values. This ultimately falls foul of Cantor's proof that the power set is strictly larger than the set." Michael further asserts that, in theory it's possible to have an constructor of type "`num -> (`α`, `β`) CCS`", in which the sub-type `num` can be replaced to γ `ordinal` to support injection from ordinals to a set of CCS processes, then the type variable of ordinals becomes part of the CCS datatype, e.g. `('a, 'b, 'c) CCS`. However, due to limitations in HOL's ordinal theory, we can't further use the CCS type to prove the existence of an ordinal which is larger than the cardinality of the set of all CCS processes.

$\vdash \exists f.\ \forall rs\ u\ E.\ f\ rs$ `--`u`->` $E \iff \exists r.\ r \in rs \land r$ `--`u`->` E

Above axiom simply asserts the existence of an infinite sum of CCS processes, and all its transitions come from the transition of any process in the set. With above axiom, we can then define the infinite summ operator (through its behavior):

Definition 15. $\vdash$ `summ` rs `--`u`->` $E \iff \exists r.\ r \in rs \land r$ `--`u`->` E

Now we can define the full version of Klop function based on ordinals:

Definition 16. *(Klop function in HOL4, full version)*

$$
\begin{array}{l}
\texttt{Klop}\ a\ 0o\ \texttt{= nil} \\
\texttt{Klop}\ a\ \alpha^{+}\ \texttt{= Klop}\ a\ \alpha\ \texttt{+ label}\ a\texttt{..Klop}\ a\ \alpha \\
0o < \alpha \land \texttt{islimit}\ \alpha \Rightarrow \\
\texttt{Klop}\ a\ \alpha\ \texttt{= summ (IMAGE (Klop}\ a\texttt{) (preds}\ \alpha\texttt{))}
\end{array}
$$

Using above definition, we can further prove the following "cases" theorem for possible transitions of infinite sums:

Proposition 21. *("cases" theorems for transitions of Klop processes)*

$$
\begin{array}{l}
\vdash (\forall a.\ \texttt{Klop}\ a\ 0o\ \texttt{= nil}) \land \\
\quad (\forall a\ n\ u\ E. \\
\qquad \texttt{Klop}\ a\ n^{+}\ \texttt{--}u\texttt{->}\ E \iff \\
\qquad u\ \texttt{= label}\ a \land E\ \texttt{= Klop}\ a\ n \lor \texttt{Klop}\ a\ n\ \texttt{--}u\texttt{->}\ E) \land \\
\quad \forall a\ n\ u\ E. \\
\qquad 0o < n \land \texttt{islimit}\ n \Rightarrow \\
\qquad (\texttt{Klop}\ a\ n\ \texttt{--}u\texttt{->}\ E \iff \exists m.\ m < n \land \texttt{Klop}\ a\ m\ \texttt{--}u\texttt{->}\ E)
\end{array}
$$

We can also converted them into the following inference rules for transitions which are easier for use:[21]

Proposition 22. *("rules" theorems for transitions of Klop processes)*

$$
\begin{array}{l}
\vdash (\forall a\ n.\ \texttt{Klop}\ a\ n^{+}\ \texttt{--label}\ a\texttt{->}\ \texttt{Klop}\ a\ n) \land \\
\quad \forall a\ n\ m\ u\ E. \\
\qquad 0o < n \land \texttt{islimit}\ n \land m < n \land \texttt{Klop}\ a\ m\ \texttt{--}u\texttt{->}\ E \Rightarrow \\
\qquad \texttt{Klop}\ a\ n\ \texttt{--}u\texttt{->}\ E
\end{array}
$$

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 of Klop processes:

$\vdash \texttt{STABLE (Klop}\ a\ n\texttt{)}$	[Klop_PROP0]
$\vdash \texttt{Klop}\ a\ n\ \texttt{--label}\ a\texttt{->}\ E \iff \exists m.\ m < n \land E\ \texttt{= Klop}\ a\ m$	[Klop_PROP1]
$\vdash \texttt{Klop}\ a\ n\ \texttt{==label}\ a\texttt{=>>}\ E \iff \exists m.\ m < n \land E\ \texttt{= Klop}\ a\ m$	[Klop_PROP1']
$\vdash m < n \Rightarrow \neg(\texttt{Klop}\ a\ m \sim \texttt{Klop}\ a\ n)$	[Klop_PROP2]
$\vdash m < n \Rightarrow \neg(\texttt{Klop}\ a\ m \approx \texttt{Klop}\ a\ n)$	[Klop_PROP2']
$\vdash \texttt{ONE\_ONE (Klop}\ a\texttt{)}$	[Klop_ONE_ONE]

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

$\vdash (\forall min.\ (\forall b.\ b < min \Rightarrow P\ b) \Rightarrow P\ min) \Rightarrow \forall \alpha.\ P\ \alpha$	[ord_inductition]
$\vdash P\ 0 \land (\forall \alpha.\ P\ \alpha \Rightarrow P\ \alpha^{+}) \land$ $(\forall \alpha.\ \texttt{islimit}\ \alpha \land 0 < \alpha \land (\forall \beta.\ \beta < \alpha \Rightarrow P\ \beta) \Rightarrow P\ \alpha) \Rightarrow$ $\forall \alpha.\ P\ \alpha$	[simple_ord_induction]

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:

[21] 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

Theorem 17. *For any arbitrary set of CCS processes, it's always possible to find a Klop process which is not weakly bisimilar with any process in the set:*

$\vdash \forall a\ A.\ \exists n.\ \forall x.\ x \in A \Rightarrow \neg(x \approx \texttt{Klop}\ a\ n)$

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

$\vdash \mathcal{U}(:\alpha\ \texttt{inf}) \prec \mathcal{U}(:\alpha\ \texttt{ordinal})$ [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, our formal proofs of mathematics theorems is not based on *Zermelo–Fraenkel (ZF)* or *von Neumann–Bernays–Gödel (NBG)* set theory but a special set theory in higher-order logic. 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 p and q:

$$f(n) = \begin{cases} n & \text{if } n < \omega, \\ \{y\colon y \in B \wedge y \approx Klop_n\} & \text{if } n \geqslant \omega. \end{cases} \tag{2}$$

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-theory stuff removed, is the following "existence" theorem:

Theorem 18. *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 $f(n) \notin A$.*

$\vdash \forall (A :\alpha \texttt{ -> bool})\ (f :\alpha\ \texttt{ordinal -> }\alpha).$
$\quad \texttt{ONE\_ONE}\ f \Rightarrow \exists (n :\alpha\ \texttt{ordinal}).\ f\ n \notin A$

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.[22]

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

Lemma 8. *(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:*

$\vdash \forall p\ q.$
$\quad \exists k.$
$\qquad \texttt{STABLE}\ k \wedge (\forall p'\ u.\ p \texttt{ ==}u\texttt{=>> } p' \Rightarrow \neg(p' \approx k)) \wedge$
$\qquad \forall q'\ u.\ q \texttt{ ==}u\texttt{=>> } q' \Rightarrow \neg(q' \approx k)$ [KLOP_LEMMA]

[22] Above theorem also indicates 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.

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 19. *(Coarsest congruence contained in $\approx$, the final version) For any processes p and q, $p \approx^c q$ if and only if $p + r \approx q + r$ for all processes r:*

$\vdash \forall p\ q.\ p \approx^c q \iff \forall r.\ p + r \approx q + r$ [COARSEST_CONGR_FULL]

Going back to the congruence theory presented in previous section. Now we can conclude that, the three relations (observation congruence, weak bisimulation congruence, and the temporarily defined "sum equivalence") coincide:

Theorem 20. *(The equivalence of three relations)*

⊢ `OBS_CONGR = SUM_EQUIV`
⊢ `OBS_CONGR = WEAK_CONGR`

10 Conclusions

In this project, we have done a further formalization of the process algebra CCS in HOL4. Most results on strong equivalence, weak equivalence and observation congruence were all formally proved. A rather complete theory of congruence (for CCS) is also presented in this project.

The project began with an old formalization of CCS in Hol88 by Monica Nesi, then it's extended with formal proofs of deep lemmas and theorems, including Hennessy Lemma, Deng Lemma, and the "coarsest congruence containing in weak equivalence" theorem. We believe these work have shown the possibility to use this project as a research basis for discovering new theorems about CCS.

For the last theorem, we deeply investigated various versions of the theorem and their proofs in original papers. But for the most general case, an infinite sum of CCS processes must be used during the proof, and we had to add an axiom to assert the existence of infinite sums. Without this axiom, the best result we could get is only for finite-state CCS. The consistency of HOL logic after adding this axiom is yet to be checked. On the other side, complete removing of this axiom seems impossible in scope of higher order logic.

We have extensively used HOL's rich theories to simplify the development efforts in this project, notable ones include: `relationTheory` (for RTC) and `ordinalTheory` (for Klop function defined on ordinals). Now we use HOL's built-in co-inductive relation support to define strong and weak equivalence, as the result many intermediate results were not needed thus removed from the old scripts.

Some missing pieces include: the decision procedures for bisimilarity checking (strong, weak and rooted week), HML and example models. For these missing pieces, a further project is already in plan.

Thanks to Prof. Roberto Gorrieri, who taught CCS and LTS theory to the author, and his supports on continuing this HOL-CCS project as exam project of his course.

Thanks to Prof. Monica Nesi for finding and sending the old HOL88 proof scripts to the author.

Thanks to Prof. Andrea Asperti, who taught the interactive theorem proving techniques to the author, although it's in another different theorem prover (Matita).

Thanks to people from HOL community (Thomas Tuerk, Michael Norrish, Ramana Kumar and many others) for resolving issues and doubts that the author met when using HOL theorem prover.

The paper is written in LaTeXand LNCS template, with theorems generated automatically by HOL's TeXexporting module (`EmitTex`) from the proof scripts.

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.: Formalising a Value-Passing Calculus in HOL. Formal Aspects of Computing **11**(2) (1999) 160–199
3. Gorrieri, R.: CCS(25, 12) is Turing-complete. submitted for publication
4. Gorrieri, R., Versari, C.: Introduction to Concurrency Theory. Transition Systems and CCS. Springer, Cham (September 2015)
5. Milner, R.: Communication and concurrency. Prentice Hall (1989)
6. van Glabbeek, R.J.: A characterisation of weak bisimulation congruence. Lecture notes in computer science **3838** (2005) 26–39
7. Nesi, M.: A formalization of the process algebra CCS in high order logic. Technical report, University of Cambridge, Computer Laboratory (1992)