Contents

1 CCS Theory **3**
- 1.1 Datatypes . . . 3
- 1.2 Definitions . . . 3
- 1.3 Theorems . . . 5

2 StrongEQ Theory **15**
- 2.1 Definitions . . . 15
- 2.2 Theorems . . . 16

3 StrongLaws Theory **19**
- 3.1 Definitions . . . 19
- 3.2 Theorems . . . 20

4 WeakEQ Theory **26**
- 4.1 Definitions . . . 26
- 4.2 Theorems . . . 28

5 Example Theory **32**
- 5.1 Theorems . . . 32

1 CCS Theory

Built: 14 Maggio 2017
Parent Theories: string

1.1 Datatypes

```
Action = tau | label Label

CCS =
    nil
  | var string
  | prefix Action CCS
  | (+) CCS CCS
  | par CCS CCS
  | nu (Label -> bool) CCS
  | relab CCS Relabeling
  | rec string CCS

Label = name string | coname string
```

1.2 Definitions

[Apply_Relab_def]

```
⊢ (∀l. Apply_Relab [] l = l) ∧
  ∀newold ls l.
    Apply_Relab (newold::ls) l =
    if SND newold = l then FST newold
    else if COMPL (SND newold) = l then COMPL (FST newold)
    else Apply_Relab ls l
```

[ARB'_def]

```
⊢ ARB' = εx. T
```

[CCS_Subst_def]

```
⊢ (∀E' X. CCS_Subst nil E' X = nil) ∧
  (∀u E E' X. CCS_Subst (u..E) E' X = u..CCS_Subst E E' X) ∧
  (∀E1 E2 E' X.
     CCS_Subst (E1 + E2) E' X =
     CCS_Subst E1 E' X + CCS_Subst E2 E' X) ∧
  (∀E1 E2 E' X.
     CCS_Subst (E1 || E2) E' X =
     CCS_Subst E1 E' X || CCS_Subst E2 E' X) ∧
  (∀L E E' X.
     CCS_Subst (nu L E) E' X = nu L (CCS_Subst E E' X)) ∧
  (∀E f E' X.
     CCS_Subst (relab E f) E' X = relab (CCS_Subst E E' X) f) ∧
  (∀Y E' X.
```

```
    CCS_Subst (var Y) E' X = if Y = X then E' else var Y) ∧
  ∀ Y E E' X.
   CCS_Subst (rec Y E) E' X =
   if Y = X then rec Y E else rec Y (CCS_Subst E E' X)
```

[COMPL_ACT_def]

```
⊢ (∀ l. COMPL (label l) = label (COMPL l)) ∧ (COMPL tau = tau)
```

[COMPL_LAB_def]

```
⊢ (∀ s. COMPL (name s) = coname s) ∧
  ∀ s. COMPL (coname s) = name s
```

[In_def]

```
⊢ ∀ act. In act = label (name act)
```

[IS_LABEL_def]

```
⊢ (∀ l. IS_LABEL (label l) ⟺ T) ∧ (IS_LABEL tau ⟺ F)
```

[Is_Relabeling_def]

```
⊢ ∀ f. Is_Relabeling f ⟺ ∀ s. f (coname s) = COMPL (f (name s))
```

[LABEL_def]

```
⊢ ∀ l. LABEL (label l) = l
```

[Out_def]

```
⊢ ∀ act. Out act = label (coname act)
```

[RELAB_def]

```
⊢ ∀ labl. RELAB labl = ABS_Relabeling (Apply_Relab labl)
```

[relabel_def]

```
⊢ (∀ rf. relabel rf tau = tau) ∧
  ∀ rf l. relabel rf (label l) = label (REP_Relabeling rf l)
```

[Relabeling_ISO_DEF]

```
⊢ (∀ a. ABS_Relabeling (REP_Relabeling a) = a) ∧
  ∀ r.
    Is_Relabeling r ⟺ (REP_Relabeling (ABS_Relabeling r) = r)
```

[Relabeling_TY_DEF]

```
⊢ ∃ rep. TYPE_DEFINITION Is_Relabeling rep
```

[Restr_def]

```
⊢ ∀ n P. nu n P = nu {name n} P
```

[TRANS_def]

$$
\begin{array}{l}
\vdash \texttt{TRANS} = \\
\quad (\lambda\, a_0\ a_1\ a_2 . \\
\quad\quad \forall\, \mathit{TRANS}' . \\
\quad\quad\quad (\forall\, a_0\ a_1\ a_2 . \\
\quad\quad\quad\quad (a_0 = a_1 \texttt{..} a_2) \lor \\
\quad\quad\quad\quad (\exists\, E\ E' .\ (a_0 = E \texttt{ + } E') \land \mathit{TRANS}'\ E\ a_1\ a_2) \lor \\
\quad\quad\quad\quad (\exists\, E\ E' .\ (a_0 = E' \texttt{ + } E) \land \mathit{TRANS}'\ E\ a_1\ a_2) \lor \\
\quad\quad\quad\quad (\exists\, E\ E_1\ E' . \\
\quad\quad\quad\quad\quad (a_0 = E \texttt{ || } E') \land (a_2 = E_1 \texttt{ || } E') \land \\
\quad\quad\quad\quad\quad \mathit{TRANS}'\ E\ a_1\ E_1) \lor \\
\quad\quad\quad\quad (\exists\, E\ E_1\ E' . \\
\quad\quad\quad\quad\quad (a_0 = E' \texttt{ || } E) \land (a_2 = E' \texttt{ || } E_1) \land \\
\quad\quad\quad\quad\quad \mathit{TRANS}'\ E\ a_1\ E_1) \lor \\
\quad\quad\quad\quad (\exists\, E\ l\ E_1\ E'\ E_2 . \\
\quad\quad\quad\quad\quad (a_0 = E \texttt{ || } E') \land (a_1 = \texttt{tau}) \land (a_2 = E_1 \texttt{ || } E_2) \land \\
\quad\quad\quad\quad\quad \mathit{TRANS}'\ E\ (\texttt{label}\ l)\ E_1 \land \\
\quad\quad\quad\quad\quad \mathit{TRANS}'\ E'\ (\texttt{label}\ (\texttt{COMPL}\ l))\ E_2) \lor \\
\quad\quad\quad\quad (\exists\, E\ E'\ l\ L . \\
\quad\quad\quad\quad\quad (a_0 = \texttt{nu}\ L\ E) \land (a_2 = \texttt{nu}\ L\ E') \land \mathit{TRANS}'\ E\ a_1\ E' \land \\
\quad\quad\quad\quad\quad ((a_1 = \texttt{tau}) \lor \\
\quad\quad\quad\quad\quad\ (a_1 = \texttt{label}\ l) \land l \notin L \land \texttt{COMPL}\ l \notin L)) \lor \\
\quad\quad\quad\quad (\exists\, E\ u\ E'\ \mathit{rf} . \\
\quad\quad\quad\quad\quad (a_0 = \texttt{relab}\ E\ \mathit{rf}) \land (a_1 = \texttt{relabel}\ \mathit{rf}\ u) \land \\
\quad\quad\quad\quad\quad (a_2 = \texttt{relab}\ E'\ \mathit{rf}) \land \mathit{TRANS}'\ E\ u\ E') \lor \\
\quad\quad\quad\quad (\exists\, E\ X . \\
\quad\quad\quad\quad\quad (a_0 = \texttt{rec}\ X\ E) \land \\
\quad\quad\quad\quad\quad \mathit{TRANS}'\ (\texttt{CCS\_Subst}\ E\ (\texttt{rec}\ X\ E)\ X)\ a_1\ a_2) \Rightarrow \\
\quad\quad\quad\quad \mathit{TRANS}'\ a_0\ a_1\ a_2) \Rightarrow \\
\quad\quad\quad \mathit{TRANS}'\ a_0\ a_1\ a_2)
\end{array}
$$

1.3 Theorems

[Action_distinct_label]

$$\vdash \forall\, a.\ \texttt{label}\ a \neq \texttt{tau}$$

[Action_not_tau_is_Label]

$$\vdash \forall\, A.\ A \neq \texttt{tau} \Rightarrow \exists\, L.\ A = \texttt{label}\ L$$

[Apply_Relab_COMPL_THM]

$$
\begin{array}{l}
\vdash \forall\, \mathit{labl}\ s . \\
\quad\quad \texttt{Apply\_Relab}\ \mathit{labl}\ (\texttt{coname}\ s) = \\
\quad\quad \texttt{COMPL}\ (\texttt{Apply\_Relab}\ \mathit{labl}\ (\texttt{name}\ s))
\end{array}
$$

[APPLY_RELAB_THM]

$$
\begin{array}{l}
\vdash \forall\, \mathit{labl}'\ \mathit{labl} . \\
\quad\quad (\texttt{RELAB}\ \mathit{labl}' = \texttt{RELAB}\ \mathit{labl}) \iff \\
\quad\quad (\texttt{Apply\_Relab}\ \mathit{labl}' = \texttt{Apply\_Relab}\ \mathit{labl})
\end{array}
$$

[CCS_COND_CLAUSES]

$\vdash \forall t_1\ t_2.$
$\quad ((\textbf{if}\ \mathtt{T}\ \textbf{then}\ t_1\ \textbf{else}\ t_2) = t_1) \land$
$\quad ((\textbf{if}\ \mathtt{F}\ \textbf{then}\ t_1\ \textbf{else}\ t_2) = t_2)$

[CCS_distinct']

$\vdash (\forall a.\ \mathtt{nil} \neq \mathtt{var}\ a) \land (\forall a_1\ a_0.\ \mathtt{nil} \neq a_0..a_1) \land$
$(\forall a_1\ a_0.\ \mathtt{nil} \neq a_0 + a_1) \land (\forall a_1\ a_0.\ \mathtt{nil} \neq a_0 \mid\mid a_1) \land$
$(\forall a_1\ a_0.\ \mathtt{nil} \neq \mathtt{nu}\ a_0\ a_1) \land (\forall a_1\ a_0.\ \mathtt{nil} \neq \mathtt{relab}\ a_0\ a_1) \land$
$(\forall a_1\ a_0.\ \mathtt{nil} \neq \mathtt{rec}\ a_0\ a_1) \land (\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq a_0..a_1) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq a_0 + a_1) \land (\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq a_0 \mid\mid a_1) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq \mathtt{nu}\ a_0\ a_1) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq \mathtt{relab}\ a_0\ a_1) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{var}\ a \neq \mathtt{rec}\ a_0\ a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0..a_1 \neq a_0' + a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0..a_1 \neq a_0' \mid\mid a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0..a_1 \neq \mathtt{nu}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0..a_1 \neq \mathtt{relab}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0..a_1 \neq \mathtt{rec}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 + a_1 \neq a_0' \mid\mid a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 + a_1 \neq \mathtt{nu}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 + a_1 \neq \mathtt{relab}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 + a_1 \neq \mathtt{rec}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 \mid\mid a_1 \neq \mathtt{nu}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 \mid\mid a_1 \neq \mathtt{relab}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0 \mid\mid a_1 \neq \mathtt{rec}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{nu}\ a_0\ a_1 \neq \mathtt{relab}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{nu}\ a_0\ a_1 \neq \mathtt{rec}\ a_0'\ a_1') \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{relab}\ a_0\ a_1 \neq \mathtt{rec}\ a_0'\ a_1') \land$
$(\forall a.\ \mathtt{var}\ a \neq \mathtt{nil}) \land (\forall a_1\ a_0.\ a_0..a_1 \neq \mathtt{nil}) \land$
$(\forall a_1\ a_0.\ a_0 + a_1 \neq \mathtt{nil}) \land (\forall a_1\ a_0.\ a_0 \mid\mid a_1 \neq \mathtt{nil}) \land$
$(\forall a_1\ a_0.\ \mathtt{nu}\ a_0\ a_1 \neq \mathtt{nil}) \land (\forall a_1\ a_0.\ \mathtt{relab}\ a_0\ a_1 \neq \mathtt{nil}) \land$
$(\forall a_1\ a_0.\ \mathtt{rec}\ a_0\ a_1 \neq \mathtt{nil}) \land (\forall a_1\ a_0\ a.\ a_0..a_1 \neq \mathtt{var}\ a) \land$
$(\forall a_1\ a_0\ a.\ a_0 + a_1 \neq \mathtt{var}\ a) \land (\forall a_1\ a_0\ a.\ a_0 \mid\mid a_1 \neq \mathtt{var}\ a) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{nu}\ a_0\ a_1 \neq \mathtt{var}\ a) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{relab}\ a_0\ a_1 \neq \mathtt{var}\ a) \land$
$(\forall a_1\ a_0\ a.\ \mathtt{rec}\ a_0\ a_1 \neq \mathtt{var}\ a) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0' + a_1' \neq a_0..a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0' \mid\mid a_1' \neq a_0..a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{nu}\ a_0'\ a_1' \neq a_0..a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{relab}\ a_0'\ a_1' \neq a_0..a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{rec}\ a_0'\ a_1' \neq a_0..a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ a_0' \mid\mid a_1' \neq a_0 + a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{nu}\ a_0'\ a_1' \neq a_0 + a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{relab}\ a_0'\ a_1' \neq a_0 + a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{rec}\ a_0'\ a_1' \neq a_0 + a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{nu}\ a_0'\ a_1' \neq a_0 \mid\mid a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{relab}\ a_0'\ a_1' \neq a_0 \mid\mid a_1) \land$
$(\forall a_1'\ a_1\ a_0'\ a_0.\ \mathtt{rec}\ a_0'\ a_1' \neq a_0 \mid\mid a_1) \land$

$(\forall a_1'\ a_1\ a_0'\ a_0.$ `relab` $a_0'\ a_1' \neq$ `nu` $a_0\ a_1) \wedge$
$(\forall a_1'\ a_1\ a_0'\ a_0.$ `rec` $a_0'\ a_1' \neq$ `nu` $a_0\ a_1) \wedge$
$\forall a_1'\ a_1\ a_0'\ a_0.$ `rec` $a_0'\ a_1' \neq$ `relab` $a_0\ a_1$

[COMPL_COMPL_ACT]

$\vdash \forall a.$ `COMPL (COMPL` a`) =` a

[COMPL_COMPL_LAB]

$\vdash \forall l.$ `COMPL (COMPL` l`) =` l

[COMPL_THM]

$\vdash \forall l\ s.$
  $(l \neq$ `name` $s \Rightarrow$ `COMPL` $l \neq$ `coname` $s) \wedge$
  $(l \neq$ `coname` $s \Rightarrow$ `COMPL` $l \neq$ `name` $s)$

[coname_COMPL]

$\vdash \forall s.$ `coname` s `= COMPL (name` s`)`

[EXISTS_Relabeling]

$\vdash \exists f.$ `Is_Relabeling` f

[IS_RELABELING]

$\vdash \forall labl.$ `Is_Relabeling (Apply_Relab` $labl$`)`

[Label_distinct']

$\vdash \forall a'\ a.$ `coname` $a' \neq$ `name` a

[Label_not_eq]

$\vdash \forall a'\ a.$ `(name` a `= coname` a'`)` $\iff$ `F`

[Label_not_eq']

$\vdash \forall a'\ a.$ `(coname` a' `= name` a`)` $\iff$ `F`

[NIL_NO_TRANS]

$\vdash \forall u\ E.\ \neg$`(nil --`$u$`->` E`)`

[NIL_NO_TRANS_EQF]

$\vdash \forall u\ E.$ `nil --`u`->` $E \iff$ `F`

[PAR1]

$\vdash \forall E\ u\ E_1\ E'.\ E$ `--`u`->` $E_1 \Rightarrow E$ `||` E' `--`u`->` E_1 `||` E'

[PAR2]

$\vdash \forall E\ u\ E_1\ E'.\ E$ `--`u`->` $E_1 \Rightarrow E'$ `||` E `--`u`->` E' `||` E_1

[PAR3]

$\vdash \forall E\ l\ E_1\ E'\ E_2.$
$\quad E$ `--label` l`->` $E_1 \land E'$ `--label (COMPL` l`)->` $E_2 \Rightarrow$
$\quad E$ `||` E' `--tau->` E_1 `||` E_2

[PAR_cases]

$\vdash \forall D\ D'\ u\ D''.$
$\quad D$ `||` D' `--`u`->` $D'' \Rightarrow$
$\quad (\exists E\ E_1\ E'.$
$\qquad ((D = E) \land (D' = E')) \land (D'' = E_1$ `||` $E') \land E$ `--`u`->` $E_1) \lor$
$\quad (\exists E\ E_1\ E'.$
$\qquad ((D = E') \land (D' = E)) \land (D'' = E'$ `||` $E_1) \land E$ `--`u`->` $E_1) \lor$
$\quad \exists E\ l\ E_1\ E'\ E_2.$
$\qquad ((D = E) \land (D' = E')) \land (u =$ `tau`$) \land (D'' = E_1$ `||` $E_2) \land$
$\qquad E$ `--label` l`->` $E_1 \land E'$ `--label (COMPL` l`)->` E_2

[PAR_cases_EQ]

$\vdash \forall D\ D'\ u\ D''.$
$\quad D$ `||` D' `--`u`->` $D'' \iff$
$\quad (\exists E\ E_1\ E'.$
$\qquad ((D = E) \land (D' = E')) \land (D'' = E_1$ `||` $E') \land E$ `--`u`->` $E_1) \lor$
$\quad (\exists E\ E_1\ E'.$
$\qquad ((D = E') \land (D' = E)) \land (D'' = E'$ `||` $E_1) \land E$ `--`u`->` $E_1) \lor$
$\quad \exists E\ l\ E_1\ E'\ E_2.$
$\qquad ((D = E) \land (D' = E')) \land (u =$ `tau`$) \land (D'' = E_1$ `||` $E_2) \land$
$\qquad E$ `--label` l`->` $E_1 \land E'$ `--label (COMPL` l`)->` E_2

[PREFIX]

$\vdash \forall E\ u.\ u..E$ `--`u`->` E

[REC]

$\vdash \forall E\ u\ X\ E_1.$
$\quad$ `CCS_Subst` E `(rec` $X\ E$`)` X `--`u`->` $E_1 \Rightarrow$ `rec` $X\ E$ `--`u`->` E_1

[REC_cases]

$\vdash \forall X\ E\ u\ E''.$
$\quad$ `rec` $X\ E$ `--`u`->` $E'' \Rightarrow$
$\quad \exists E'\ X'.$
$\qquad ((X = X') \land (E = E')) \land$
$\qquad$ `CCS_Subst` E' `(rec` $X'\ E'$`)` X' `--`u`->` E''

[REC_cases_EQ]

$\vdash \forall X\ E\ u\ E''.$
$\quad$ `rec` $X\ E$ `--`u`->` $E'' \iff$
$\quad \exists E'\ X'.$
$\qquad ((X = X') \land (E = E')) \land$
$\qquad$ `CCS_Subst` E' `(rec` $X'\ E'$`)` X' `--`u`->` E''

[RELAB]

$\vdash \forall E\ u\ E'\ rf.$
$\quad E$ `--`u`->` $E' \Rightarrow$ `relab` $E\ rf$ `--relabel` $rf\ u$`->` `relab` $E'\ rf$

[RELAB_cases]

$\vdash \forall rf\ E\ a_1\ a_2.$
$\quad$ `relab` $E\ rf$ `--`a_1`->` $a_2 \Rightarrow$
$\quad \exists E'\ u\ E''\ rf'.$
$\qquad ((E = E') \wedge (rf = rf')) \wedge (a_1 =$ `relabel` $rf'\ u) \wedge$
$\qquad (a_2 =$ `relab` $E''\ rf') \wedge E'$ `--`u`->` E''

[RELAB_cases_EQ]

$\vdash \forall rf\ E\ a_1\ a_2.$
$\quad$ `relab` $E\ rf$ `--`a_1`->` $a_2 \iff$
$\quad \exists E'\ u\ E''\ rf'.$
$\qquad ((E = E') \wedge (rf = rf')) \wedge (a_1 =$ `relabel` $rf'\ u) \wedge$
$\qquad (a_2 =$ `relab` $E''\ rf') \wedge E'$ `--`u`->` E''

[RELAB_NIL_NO_TRANS]

$\vdash \forall rf\ u\ E.\ \neg($`relab nil` rf `--`u`->` $E)$

[REP_Relabeling_THM]

$\vdash \forall rf.$ `Is_Relabeling (REP_Relabeling` rf`)`

[RESTR]

$\vdash \forall E\ u\ E'\ l\ L.$
$\quad E$ `--`u`->` $E' \wedge$
$\quad ((u =$ `tau`$) \vee (u =$ `label` $l) \wedge l \notin L \wedge$ `COMPL` $l \notin L) \Rightarrow$
$\quad$ `nu` $L\ E$ `--`u`->` `nu` $L\ E'$

[RESTR_cases]

$\vdash \forall D\ L\ u\ D'.$
$\quad$ `nu` $L\ D$ `--`u`->` $D' \Rightarrow$
$\quad \exists E\ E'\ l\ L'.$
$\qquad ((L = L') \wedge (D = E)) \wedge (D' =$ `nu` $L'\ E') \wedge E$ `--`u`->` $E' \wedge$
$\qquad ((u =$ `tau`$) \vee (u =$ `label` $l) \wedge l \notin L' \wedge$ `COMPL` $l \notin L')$

[RESTR_cases_EQ]

$\vdash \forall D\ L\ u\ D'.$
$\quad$ `nu` $L\ D$ `--`u`->` $D' \iff$
$\quad \exists E\ E'\ l\ L'.$
$\qquad ((L = L') \wedge (D = E)) \wedge (D' =$ `nu` $L'\ E') \wedge E$ `--`u`->` $E' \wedge$
$\qquad ((u =$ `tau`$) \vee (u =$ `label` $l) \wedge l \notin L' \wedge$ `COMPL` $l \notin L')$

[RESTR_LABEL_NO_TRANS]

$\vdash \forall l\ L.$
$\quad l \in L \vee$ `COMPL` $l \in L \Rightarrow$
$\quad \forall E\ u\ E'.\ \neg($`nu` L `(label` l`..`E`)` `--`u`->` $E')$

[RESTR_NIL_NO_TRANS]

$\vdash \forall L\ u\ E.\ \neg(\texttt{nu}\ L\ \texttt{nil}\ \texttt{--}u\texttt{->}\ E)$

[SUM1]

$\vdash \forall E\ u\ E_1\ E'.\ E\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow E\ \texttt{+}\ E'\ \texttt{--}u\texttt{->}\ E_1$

[SUM2]

$\vdash \forall E\ u\ E_1\ E'.\ E\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow E'\ \texttt{+}\ E\ \texttt{--}u\texttt{->}\ E_1$

[SUM_cases]

$$
\begin{array}{l}
\vdash \forall D\ D'\ u\ D''. \\
\quad D\ \texttt{+}\ D'\ \texttt{--}u\texttt{->}\ D'' \Rightarrow \\
\quad (\exists E\ E'.\ ((D = E) \land (D' = E')) \land E\ \texttt{--}u\texttt{->}\ D'') \lor \\
\quad \exists E\ E'.\ ((D = E') \land (D' = E)) \land E\ \texttt{--}u\texttt{->}\ D''
\end{array}
$$

[SUM_cases_EQ]

$$
\begin{array}{l}
\vdash \forall D\ D'\ u\ D''. \\
\quad D\ \texttt{+}\ D'\ \texttt{--}u\texttt{->}\ D'' \iff \\
\quad (\exists E\ E'.\ ((D = E) \land (D' = E')) \land E\ \texttt{--}u\texttt{->}\ D'') \lor \\
\quad \exists E\ E'.\ ((D = E') \land (D' = E)) \land E\ \texttt{--}u\texttt{->}\ D''
\end{array}
$$

[TRANS_ASSOC_EQ]

$$
\begin{array}{l}
\vdash \forall E\ E'\ E''\ E_1\ u. \\
\quad E\ \texttt{+}\ E'\ \texttt{+}\ E''\ \texttt{--}u\texttt{->}\ E_1 \iff E\ \texttt{+}\ (E'\ \texttt{+}\ E'')\ \texttt{--}u\texttt{->}\ E_1
\end{array}
$$

[TRANS_ASSOC_RL]

$$
\begin{array}{l}
\vdash \forall E\ E'\ E''\ E_1\ u. \\
\quad E\ \texttt{+}\ (E'\ \texttt{+}\ E'')\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow E\ \texttt{+}\ E'\ \texttt{+}\ E''\ \texttt{--}u\texttt{->}\ E_1
\end{array}
$$

[TRANS_cases]

$$
\begin{array}{l}
\vdash \forall a_0\ a_1\ a_2. \\
\quad a_0\ \texttt{--}a_1\texttt{->}\ a_2 \iff \\
\quad (a_0 = a_1\texttt{..}a_2) \lor (\exists E\ E'.\ (a_0 = E\ \texttt{+}\ E') \land E\ \texttt{--}a_1\texttt{->}\ a_2) \lor \\
\quad (\exists E\ E'.\ (a_0 = E'\ \texttt{+}\ E) \land E\ \texttt{--}a_1\texttt{->}\ a_2) \lor \\
\quad (\exists E\ E_1\ E'. \\
\quad\quad (a_0 = E\ \texttt{||}\ E') \land (a_2 = E_1\ \texttt{||}\ E') \land E\ \texttt{--}a_1\texttt{->}\ E_1) \lor \\
\quad (\exists E\ E_1\ E'. \\
\quad\quad (a_0 = E'\ \texttt{||}\ E) \land (a_2 = E'\ \texttt{||}\ E_1) \land E\ \texttt{--}a_1\texttt{->}\ E_1) \lor \\
\quad (\exists E\ l\ E_1\ E'\ E_2. \\
\quad\quad (a_0 = E\ \texttt{||}\ E') \land (a_1 = \texttt{tau}) \land (a_2 = E_1\ \texttt{||}\ E_2) \land \\
\quad\quad E\ \texttt{--label}\ l\texttt{->}\ E_1 \land E'\ \texttt{--label (COMPL}\ l\texttt{)->}\ E_2) \lor \\
\quad (\exists E\ E'\ l\ L. \\
\quad\quad (a_0 = \texttt{nu}\ L\ E) \land (a_2 = \texttt{nu}\ L\ E') \land E\ \texttt{--}a_1\texttt{->}\ E' \land \\
\quad\quad ((a_1 = \texttt{tau}) \lor (a_1 = \texttt{label}\ l) \land l \notin L \land \texttt{COMPL}\ l \notin L)) \lor \\
\quad (\exists E\ u\ E'\ rf. \\
\quad\quad (a_0 = \texttt{relab}\ E\ rf) \land (a_1 = \texttt{relabel}\ rf\ u) \land \\
\quad\quad (a_2 = \texttt{relab}\ E'\ rf) \land E\ \texttt{--}u\texttt{->}\ E') \lor \\
\quad \exists E\ X.\ (a_0 = \texttt{rec}\ X\ E) \land \texttt{CCS\_Subst}\ E\ (\texttt{rec}\ X\ E)\ X\ \texttt{--}a_1\texttt{->}\ a_2
\end{array}
$$

[TRANS_COMM_EQ]

$\vdash \forall E\ E'\ E''\ u.\ E$ `+` E' `--`u`->` $E'' \iff E'$ `+` E `--`u`->` E''

[TRANS_IMP_NO_NIL]

$\vdash \forall E\ u\ E'.\ E$ `--`u`->` $E' \Rightarrow E \neq$ `nil`

[TRANS_IMP_NO_NIL']

$\vdash \forall E\ u\ E'.\ E$ `--`u`->` $E' \Rightarrow E \neq$ `nil`

[TRANS_IMP_NO_RESTR_NIL]

$\vdash \forall E\ u\ E'.\ E$ `--`u`->` $E' \Rightarrow \forall L.\ E \neq$ `nu` L `nil`

[TRANS_ind]

$\vdash \forall \mathit{TRANS}'.$
$\quad(\forall E\ u.\ \mathit{TRANS}'\ (u..E)\ u\ E)\ \land$
$\quad(\forall E\ u\ E_1\ E'.\ \mathit{TRANS}'\ E\ u\ E_1 \Rightarrow \mathit{TRANS}'\ (E + E')\ u\ E_1)\ \land$
$\quad(\forall E\ u\ E_1\ E'.\ \mathit{TRANS}'\ E\ u\ E_1 \Rightarrow \mathit{TRANS}'\ (E' + E)\ u\ E_1)\ \land$
$\quad(\forall E\ u\ E_1\ E'.$
$\qquad\mathit{TRANS}'\ E\ u\ E_1 \Rightarrow \mathit{TRANS}'\ (E\ ||\ E')\ u\ (E_1\ ||\ E'))\ \land$
$\quad(\forall E\ u\ E_1\ E'.$
$\qquad\mathit{TRANS}'\ E\ u\ E_1 \Rightarrow \mathit{TRANS}'\ (E'\ ||\ E)\ u\ (E'\ ||\ E_1))\ \land$
$\quad(\forall E\ l\ E_1\ E'\ E_2.$
$\qquad\mathit{TRANS}'\ E\ (\mathtt{label}\ l)\ E_1 \land \mathit{TRANS}'\ E'\ (\mathtt{label}\ (\mathtt{COMPL}\ l))\ E_2 \Rightarrow$
$\qquad\mathit{TRANS}'\ (E\ ||\ E')\ \mathtt{tau}\ (E_1\ ||\ E_2))\ \land$
$\quad(\forall E\ u\ E'\ l\ L.$
$\qquad\mathit{TRANS}'\ E\ u\ E'\ \land$
$\qquad((u = \mathtt{tau}) \lor (u = \mathtt{label}\ l) \land l \notin L \land \mathtt{COMPL}\ l \notin L) \Rightarrow$
$\qquad\mathit{TRANS}'\ (\mathtt{nu}\ L\ E)\ u\ (\mathtt{nu}\ L\ E'))\ \land$
$\quad(\forall E\ u\ E'\ \mathit{rf}.$
$\qquad\mathit{TRANS}'\ E\ u\ E' \Rightarrow$
$\qquad\mathit{TRANS}'\ (\mathtt{relab}\ E\ \mathit{rf})\ (\mathtt{relabel}\ \mathit{rf}\ u)\ (\mathtt{relab}\ E'\ \mathit{rf}))\ \land$
$\quad(\forall E\ u\ X\ E_1.$
$\qquad\mathit{TRANS}'\ (\mathtt{CCS\_Subst}\ E\ (\mathtt{rec}\ X\ E)\ X)\ u\ E_1 \Rightarrow$
$\qquad\mathit{TRANS}'\ (\mathtt{rec}\ X\ E)\ u\ E_1) \Rightarrow$
$\quad\forall a_0\ a_1\ a_2.\ a_0$ `--`a_1`->` $a_2 \Rightarrow \mathit{TRANS}'\ a_0\ a_1\ a_2$

[TRANS_P_RESTR]

$\vdash \forall E\ u\ E'\ L.$ `nu` $L\ E$ `--`u`->` `nu` $L\ E' \Rightarrow E$ `--`u`->` E'

[TRANS_P_SUM_P]

$\vdash \forall E\ u\ E'.\ E$ `+` E `--`u`->` $E' \Rightarrow E$ `--`u`->` E'

[TRANS_P_SUM_P_EQ]

$\vdash \forall E\ u\ E'.\ E$ `+` E `--`u`->` $E' \iff E$ `--`u`->` E'

[TRANS_PAR]

$\vdash \forall E\ E'\ u\ E''.$
E `||` E' `--`u`->` $E'' \Rightarrow$
$(\exists E_1.\ (E'' = E_1$ `||` $E') \land E$ `--`u`->` $E_1) \lor$
$(\exists E_1.\ (E'' = E$ `||` $E_1) \land E'$ `--`u`->` $E_1) \lor$
$\exists E_1\ E_2\ l.$
$(u =$ `tau`$) \land (E'' = E_1$ `||` $E_2) \land E$ `--label` l`->` $E_1 \land$
E' `--label (COMPL` l`)->` E_2

[TRANS_PAR_EQ]

$\vdash \forall E\ E'\ u\ E''.$
E `||` E' `--`u`->` $E'' \iff$
$(\exists E_1.\ (E'' = E_1$ `||` $E') \land E$ `--`u`->` $E_1) \lor$
$(\exists E_1.\ (E'' = E$ `||` $E_1) \land E'$ `--`u`->` $E_1) \lor$
$\exists E_1\ E_2\ l.$
$(u =$ `tau`$) \land (E'' = E_1$ `||` $E_2) \land E$ `--label` l`->` $E_1 \land$
E' `--label (COMPL` l`)->` E_2

[TRANS_PAR_NO_SYNCR]

$\vdash \forall l\ l'.$
$l \neq$ `COMPL` $l' \Rightarrow$
$\forall E\ E'\ E''.\ \neg($`label` l`..`E `||` `label` l'`..`E' `--tau->` $E'')$

[TRANS_PAR_P_NIL]

$\vdash \forall E\ u\ E'.$
E `|| nil --`u`->` $E' \Rightarrow \exists E''.\ E$ `--`u`->` $E'' \land (E' = E''$ `|| nil`$)$

[TRANS_PREFIX]

$\vdash \forall u\ E\ u'\ E'.\ u$`..`$E$ `--`u'`->` $E' \Rightarrow (u' = u) \land (E' = E)$

[TRANS_PREFIX_EQ]

$\vdash \forall u\ E\ u'\ E'.\ u$`..`$E$ `--`u'`->` $E' \iff (u' = u) \land (E' = E)$

[TRANS_REC]

$\vdash \forall X\ E\ u\ E'.$
`rec` $X\ E$ `--`u`->` $E' \Rightarrow$ `CCS_Subst` E `(rec` $X\ E$`)` X `--`u`->` E'

[TRANS_REC_EQ]

$\vdash \forall X\ E\ u\ E'.$
`rec` $X\ E$ `--`u`->` $E' \iff$ `CCS_Subst` E `(rec` $X\ E$`)` X `--`u`->` E'

[TRANS_RELAB]

$\vdash \forall E\ rf\ u\ E'.$
`relab` $E\ rf$ `--`u`->` $E' \Rightarrow$
$\exists u'\ E''.$
$(u =$ `relabel` $rf\ u') \land (E' =$ `relab` $E''\ rf) \land E$ `--`u'`->` E''

[TRANS_RELAB_EQ]

```
⊢ ∀E rf u E'.
    relab E rf --u-> E' ⟺
    ∃u' E''.
      (u = relabel rf u') ∧ (E' = relab E'' rf) ∧ E --u'-> E''
```

[TRANS_RELAB_labl]

```
⊢ ∀labl E u E'.
    relab E (RELAB labl) --u-> E' ⇒
    ∃u' E''.
      (u = relabel (RELAB labl) u') ∧
      (E' = relab E'' (RELAB labl)) ∧ E --u'-> E''
```

[TRANS_RESTR]

```
⊢ ∀E L u E'.
    nu L E --u-> E' ⇒
    ∃E'' l.
      (E' = nu L E'') ∧ E --u-> E'' ∧
      ((u = tau) ∨ (u = label l) ∧ l ∉ L ∧ COMPL l ∉ L)
```

[TRANS_RESTR_EQ]

```
⊢ ∀E L u E'.
    nu L E --u-> E' ⟺
    ∃E'' l.
      (E' = nu L E'') ∧ E --u-> E'' ∧
      ((u = tau) ∨ (u = label l) ∧ l ∉ L ∧ COMPL l ∉ L)
```

[TRANS_RESTR_NO_NIL]

```
⊢ ∀E L u E'. nu L E --u-> nu L E' ⇒ E ≠ nil
```

[TRANS_rules]

```
⊢ (∀E u. u..E --u-> E) ∧
  (∀E u E1 E'. E --u-> E1 ⇒ E + E' --u-> E1) ∧
  (∀E u E1 E'. E --u-> E1 ⇒ E' + E --u-> E1) ∧
  (∀E u E1 E'. E --u-> E1 ⇒ E || E' --u-> E1 || E') ∧
  (∀E u E1 E'. E --u-> E1 ⇒ E' || E --u-> E' || E1) ∧
  (∀E l E1 E' E2.
     E --label l-> E1 ∧ E' --label (COMPL l)-> E2 ⇒
     E || E' --tau-> E1 || E2) ∧
  (∀E u E' l L.
     E --u-> E' ∧
     ((u = tau) ∨ (u = label l) ∧ l ∉ L ∧ COMPL l ∉ L) ⇒
     nu L E --u-> nu L E') ∧
  (∀E u E' rf.
     E --u-> E' ⇒ relab E rf --relabel rf u-> relab E' rf) ∧
  ∀E u X E1.
    CCS_Subst E (rec X E) X --u-> E1 ⇒ rec X E --u-> E1
```

[TRANS_strongind]

$\vdash \forall\, TRANS'.$
$\quad (\forall\, E\ u.\ TRANS'\ (u..E)\ u\ E) \land$
$\quad (\forall\, E\ u\ E_1\ E'.$
$\qquad E\ \texttt{--}u\texttt{->}\ E_1 \land TRANS'\ E\ u\ E_1 \Rightarrow TRANS'\ (E\ \texttt{+}\ E')\ u\ E_1) \land$
$\quad (\forall\, E\ u\ E_1\ E'.$
$\qquad E\ \texttt{--}u\texttt{->}\ E_1 \land TRANS'\ E\ u\ E_1 \Rightarrow TRANS'\ (E'\ \texttt{+}\ E)\ u\ E_1) \land$
$\quad (\forall\, E\ u\ E_1\ E'.$
$\qquad E\ \texttt{--}u\texttt{->}\ E_1 \land TRANS'\ E\ u\ E_1 \Rightarrow$
$\qquad TRANS'\ (E\ \texttt{||}\ E')\ u\ (E_1\ \texttt{||}\ E')) \land$
$\quad (\forall\, E\ u\ E_1\ E'.$
$\qquad E\ \texttt{--}u\texttt{->}\ E_1 \land TRANS'\ E\ u\ E_1 \Rightarrow$
$\qquad TRANS'\ (E'\ \texttt{||}\ E)\ u\ (E'\ \texttt{||}\ E_1)) \land$
$\quad (\forall\, E\ l\ E_1\ E'\ E_2.$
$\qquad E\ \texttt{--label}\ l\texttt{->}\ E_1 \land TRANS'\ E\ (\texttt{label}\ l)\ E_1 \land$
$\qquad E'\ \texttt{--label (COMPL}\ l\texttt{)->}\ E_2 \land$
$\qquad TRANS'\ E'\ (\texttt{label (COMPL}\ l\texttt{)})\ E_2 \Rightarrow$
$\qquad TRANS'\ (E\ \texttt{||}\ E')\ \texttt{tau}\ (E_1\ \texttt{||}\ E_2)) \land$
$\quad (\forall\, E\ u\ E'\ l\ L.$
$\qquad E\ \texttt{--}u\texttt{->}\ E' \land TRANS'\ E\ u\ E' \land$
$\qquad ((u\ \texttt{=}\ \texttt{tau}) \lor (u\ \texttt{=}\ \texttt{label}\ l) \land l \notin L \land \texttt{COMPL}\ l \notin L) \Rightarrow$
$\qquad TRANS'\ (\texttt{nu}\ L\ E)\ u\ (\texttt{nu}\ L\ E')) \land$
$\quad (\forall\, E\ u\ E'\ rf.$
$\qquad E\ \texttt{--}u\texttt{->}\ E' \land TRANS'\ E\ u\ E' \Rightarrow$
$\qquad TRANS'\ (\texttt{relab}\ E\ rf)\ (\texttt{relabel}\ rf\ u)\ (\texttt{relab}\ E'\ rf)) \land$
$\quad (\forall\, E\ u\ X\ E_1.$
$\qquad \texttt{CCS\_Subst}\ E\ (\texttt{rec}\ X\ E)\ X\ \texttt{--}u\texttt{->}\ E_1 \land$
$\qquad TRANS'\ (\texttt{CCS\_Subst}\ E\ (\texttt{rec}\ X\ E)\ X)\ u\ E_1 \Rightarrow$
$\qquad TRANS'\ (\texttt{rec}\ X\ E)\ u\ E_1) \Rightarrow$
$\quad \forall\, a_0\ a_1\ a_2.\ a_0\ \texttt{--}a_1\texttt{->}\ a_2 \Rightarrow TRANS'\ a_0\ a_1\ a_2$

[TRANS_SUM]

$\vdash \forall\, E\ E'\ u\ E''.\ E\ \texttt{+}\ E'\ \texttt{--}u\texttt{->}\ E'' \Rightarrow E\ \texttt{--}u\texttt{->}\ E'' \lor E'\ \texttt{--}u\texttt{->}\ E''$

[TRANS_SUM_EQ]

$\vdash \forall\, E\ E'\ u\ E''.\ E\ \texttt{+}\ E'\ \texttt{--}u\texttt{->}\ E'' \iff E\ \texttt{--}u\texttt{->}\ E'' \lor E'\ \texttt{--}u\texttt{->}\ E''$

[TRANS_SUM_EQ']

$\vdash \forall\, E_1\ E_2\ u\ E.\ E_1\ \texttt{+}\ E_2\ \texttt{--}u\texttt{->}\ E \iff E_1\ \texttt{--}u\texttt{->}\ E \lor E_2\ \texttt{--}u\texttt{->}\ E$

[TRANS_SUM_NIL]

$\vdash \forall\, E\ u\ E'.\ E\ \texttt{+}\ \texttt{nil}\ \texttt{--}u\texttt{->}\ E' \Rightarrow E\ \texttt{--}u\texttt{->}\ E'$

[TRANS_SUM_NIL_EQ]

$\vdash \forall\, E\ u\ E'.\ E\ \texttt{+}\ \texttt{nil}\ \texttt{--}u\texttt{->}\ E' \iff E\ \texttt{--}u\texttt{->}\ E'$

[VAR_NO_TRANS]

$\vdash \forall\, X\ u\ E.\ \neg(\texttt{var}\ X\ \texttt{--}u\texttt{->}\ E)$

2 StrongEQ Theory

Built: 14 Maggio 2017
Parent Theories: CCS

2.1 Definitions

[BIGUNION_BISIM_def]

```
⊢ BIGUNION_BISIM =
  CURRY (BIGUNION {UNCURRY R | STRONG_BISIM R})
```

[STRONG_BISIM]

```
⊢ ∀ Bsm.
    STRONG_BISIM Bsm ⟺
    ∀ E E'.
      Bsm E E' ⇒
      ∀ u.
        (∀ E1. E --u-> E1 ⇒ ∃ E2. E' --u-> E2 ∧ Bsm E1 E2) ∧
        ∀ E2. E' --u-> E2 ⇒ ∃ E1. E --u-> E1 ∧ Bsm E1 E2
```

[STRONG_BISIM_UPTO]

```
⊢ ∀ Bsm.
    STRONG_BISIM_UPTO Bsm ⟺
    ∀ E E'.
      Bsm E E' ⇒
      ∀ u.
        (∀ E1.
           E --u-> E1 ⇒
           ∃ E2.
             E' --u-> E2 ∧
             (STRONG_EQUIV O Bsm O STRONG_EQUIV) E1 E2) ∧
        ∀ E2.
          E' --u-> E2 ⇒
          ∃ E1.
            E --u-> E1 ∧
            (STRONG_EQUIV O Bsm O STRONG_EQUIV) E1 E2
```

[STRONG_EQ_def]

```
⊢ STRONG_EQ =
  (λ a0 a1.
     ∃ STRONG_EQ'.
       STRONG_EQ' a0 a1 ∧
       ∀ a0 a1.
         STRONG_EQ' a0 a1 ⇒
         ∀ u.
           (∀ E1.
              a0 --u-> E1 ⇒
```

$\exists E_2.\ a_1$ `--`u`->` $E_2 \wedge STRONG\_EQ'\ E_1\ E_2) \wedge$
$\forall E_2.$
a_1 `--`u`->` $E_2 \Rightarrow \exists E_1.\ a_0$ `--`u`->` $E_1 \wedge STRONG\_EQ'\ E_1\ E_2)$

[STRONG_EQUIV]

$\vdash \forall E\ E'.\ E$ `~~` $E' \iff \exists Bsm.\ Bsm\ E\ E' \wedge$ `STRONG_BISIM` Bsm

[STRONG_EQUIV']

$\vdash \forall E\ E'.$
`STRONG_EQUIV'` $E\ E' \iff$
$\forall u.$
$(\forall E_1.\ E$ `--`u`->` $E_1 \Rightarrow \exists E_2.\ E'$ `--`u`->` $E_2 \wedge E_1$ `~~` $E_2) \wedge$
$\forall E_2.\ E'$ `--`u`->` $E_2 \Rightarrow \exists E_1.\ E$ `--`u`->` $E_1 \wedge E_1$ `~~` E_2

2.2 Theorems

[COMP_STRONG_BISIM]

$\vdash \forall Bsm_1\ Bsm_2.$
`STRONG_BISIM` $Bsm_1 \wedge$ `STRONG_BISIM` $Bsm_2 \Rightarrow$
`STRONG_BISIM` $(\lambda x\ z.\ \exists y.\ Bsm_1\ x\ y \wedge Bsm_2\ y\ z)$

[CONVERSE_STRONG_BISIM]

$\vdash \forall Bsm.$ `STRONG_BISIM` $Bsm \Rightarrow$ `STRONG_BISIM` $(\lambda x\ y.\ Bsm\ y\ x)$

[EQUAL_IMP_STRONG_EQUIV]

$\vdash \forall E\ E'.\ (E = E') \Rightarrow E$ `~~` E'

[IDENTITY_STRONG_BISIM]

$\vdash$ `STRONG_BISIM` $(\lambda x\ y.\ x = y)$

[PROPERTY_STAR]

$\vdash \forall E\ E'.$
E `~~` $E' \iff$
$\forall u.$
$(\forall E_1.\ E$ `--`u`->` $E_1 \Rightarrow \exists E_2.\ E'$ `--`u`->` $E_2 \wedge E_1$ `~~` $E_2) \wedge$
$\forall E_2.\ E'$ `--`u`->` $E_2 \Rightarrow \exists E_1.\ E$ `--`u`->` $E_1 \wedge E_1$ `~~` E_2

[PROPERTY_STAR_LR]

$\vdash \forall E\ E'.$
E `~~` $E' \Rightarrow$
$\forall u.$
$(\forall E_1.\ E$ `--`u`->` $E_1 \Rightarrow \exists E_2.\ E'$ `--`u`->` $E_2 \wedge E_1$ `~~` $E_2) \wedge$
$\forall E_2.\ E'$ `--`u`->` $E_2 \Rightarrow \exists E_1.\ E$ `--`u`->` $E_1 \wedge E_1$ `~~` E_2

[STR_EQ_IMP_STR_EQUIV]

$\vdash \forall E\ E'.$ `STRONG_EQ` $E\ E' \Rightarrow E$ `~~` E'

[STR_EQ_IS_STR_BISIM]

$\vdash$ STRONG_BISIM STRONG_EQ

[STR_EQ_TO_STR_EQUIV]

$\vdash \forall E\ E'.\ \texttt{STRONG\_EQ}\ E\ E' \iff E\ \texttt{\~{}\~{}}\ E'$

[STR_EQUIV'_IMP_STR_EQUIV]

$\vdash \forall E\ E'.\ \texttt{STRONG\_EQUIV'}\ E\ E' \Rightarrow E\ \texttt{\~{}\~{}}\ E'$

[STR_EQUIV'_TO_STR_EQUIV]

$\vdash \forall E\ E'.\ \texttt{STRONG\_EQUIV'}\ E\ E' \iff E\ \texttt{\~{}\~{}}\ E'$

[STR_EQUIV_IMP_STR_EQ]

$\vdash \forall E\ E'.\ E\ \texttt{\~{}\~{}}\ E' \Rightarrow \texttt{STRONG\_EQ}\ E\ E'$

[STR_EQUIV_IMP_STR_EQUIV']

$\vdash \forall E\ E'.\ E\ \texttt{\~{}\~{}}\ E' \Rightarrow \texttt{STRONG\_EQUIV'}\ E\ E'$

[STR_EQUIV_TO_STR_EQ]

$\vdash \forall E\ E'.\ E\ \texttt{\~{}\~{}}\ E' \iff \texttt{STRONG\_EQ}\ E\ E'$

[STR_EQUIV_TO_STR_EQUIV']

$\vdash \forall E\ E'.\ E\ \texttt{\~{}\~{}}\ E' \iff \texttt{STRONG\_EQUIV'}\ E\ E'$

[STRONG_EQ_cases]

$$
\begin{array}{l}
\vdash \forall a_0\ a_1.\\
\quad \texttt{STRONG\_EQ}\ a_0\ a_1 \iff\\
\quad \forall u.\\
\quad\quad (\forall E_1.\ a_0\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow \exists E_2.\ a_1\ \texttt{--}u\texttt{->}\ E_2 \land \texttt{STRONG\_EQ}\ E_1\ E_2) \land\\
\quad\quad \forall E_2.\ a_1\ \texttt{--}u\texttt{->}\ E_2 \Rightarrow \exists E_1.\ a_0\ \texttt{--}u\texttt{->}\ E_1 \land \texttt{STRONG\_EQ}\ E_1\ E_2
\end{array}
$$

[STRONG_EQ_coind]

$$
\begin{array}{l}
\vdash \forall \mathit{STRONG\_EQ}'.\\
\quad (\forall a_0\ a_1.\\
\quad\quad \mathit{STRONG\_EQ}'\ a_0\ a_1 \Rightarrow\\
\quad\quad \forall u.\\
\quad\quad\quad (\forall E_1.\\
\quad\quad\quad\quad a_0\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow\\
\quad\quad\quad\quad \exists E_2.\ a_1\ \texttt{--}u\texttt{->}\ E_2 \land \mathit{STRONG\_EQ}'\ E_1\ E_2) \land\\
\quad\quad\quad \forall E_2.\\
\quad\quad\quad\quad a_1\ \texttt{--}u\texttt{->}\ E_2 \Rightarrow \exists E_1.\ a_0\ \texttt{--}u\texttt{->}\ E_1 \land \mathit{STRONG\_EQ}'\ E_1\ E_2) \Rightarrow\\
\quad \forall a_0\ a_1.\ \mathit{STRONG\_EQ}'\ a_0\ a_1 \Rightarrow \texttt{STRONG\_EQ}\ a_0\ a_1
\end{array}
$$

[STRONG_EQ_rules]

$$\vdash \forall E\ E'.\ (\forall u.\ (\forall E_1.\ E\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow \exists E_2.\ E'\ \texttt{--}u\texttt{->}\ E_2 \land \texttt{STRONG\_EQ}\ E_1\ E_2) \land \forall E_2.\ E'\ \texttt{--}u\texttt{->}\ E_2 \Rightarrow \exists E_1.\ E\ \texttt{--}u\texttt{->}\ E_1 \land \texttt{STRONG\_EQ}\ E_1\ E_2) \Rightarrow \texttt{STRONG\_EQ}\ E\ E'$$

[STRONG_EQUIV_coind]

$$\vdash \forall R.\ (\forall E\ E'.\ R\ E\ E' \Rightarrow \forall u.\ (\forall E_1.\ E\ \texttt{--}u\texttt{->}\ E_1 \Rightarrow \exists E_2.\ E'\ \texttt{--}u\texttt{->}\ E_2 \land R\ E_1\ E_2) \land \forall E_2.\ E'\ \texttt{--}u\texttt{->}\ E_2 \Rightarrow \exists E_1.\ E\ \texttt{--}u\texttt{->}\ E_1 \land R\ E_1\ E_2) \Rightarrow \forall E\ E'.\ R\ E\ E' \Rightarrow E\ \texttt{\~{}\~{}}\ E'$$

[STRONG_EQUIV_EQ_BIGUNION_BISIM]

$\vdash$ STRONG_EQUIV = CURRY (BIGUNION $\{$UNCURRY R | STRONG_BISIM $R\}$)

[STRONG_EQUIV_IS_BIGUNION_BISIM]

$\vdash \forall E\ E'.\ E$ ~~ $E' \iff$ BIGUNION_BISIM $E\ E'$

[STRONG_EQUIV_IS_BIGUNION_BISIM']

$\vdash$ STRONG_EQUIV = BIGUNION_BISIM

[STRONG_EQUIV_IS_BIGUNION_BISIM'']

$\vdash$ STRONG_EQUIV = BIGUNION_BISIM

[STRONG_EQUIV_IS_STRONG_BISIM]

$\vdash$ STRONG_BISIM STRONG_EQUIV'

[STRONG_EQUIV_PRESD_BY_PAR]

$\vdash \forall E_1\ E_1'\ E_2\ E_2'.$
E_1 ~~ $E_1' \land E_2$ ~~ $E_2' \Rightarrow E_1$ || E_2 ~~ E_1' || E_2'

[STRONG_EQUIV_PRESD_BY_SUM]

$\vdash \forall E_1\ E_1'\ E_2\ E_2'.\ E_1$ ~~ $E_1' \land E_2$ ~~ $E_2' \Rightarrow E_1 + E_2$ ~~ $E_1' + E_2'$

[STRONG_EQUIV_REFL]

$\vdash \forall E.\ E$ ~~ E

[STRONG_EQUIV_SUBST_PAR_L]

$\vdash \forall E'\ E.\ E$ ~~ $E' \Rightarrow \forall E''.\ E''$ || E ~~ E'' || E'

[STRONG_EQUIV_SUBST_PAR_R]

$\vdash \forall E'\ E.\ E$ ~~ $E' \Rightarrow \forall E''.\ E$ || E'' ~~ E' || E''

[STRONG_EQUIV_SUBST_PREFIX]

$\vdash \forall E\ E'.\ E$ ~~ $E' \Rightarrow \forall u.\ u..E$ ~~ $u..E'$

[STRONG_EQUIV_SUBST_RELAB]

$\vdash \forall E\ E'.\ E$ ~~ $E' \Rightarrow \forall rf.$ relab $E\ rf$ ~~ relab $E'\ rf$

[STRONG_EQUIV_SUBST_RESTR]

$\vdash \forall E\ E'.\ E$ ~~ $E' \Rightarrow \forall L.$ nu $L\ E$ ~~ nu $L\ E'$

[STRONG_EQUIV_SUBST_SUM_L]

$\vdash \forall E'\ E.\ E$ ~~ $E' \Rightarrow \forall E''.\ E'' + E$ ~~ $E'' + E'$

[STRONG_EQUIV_SUBST_SUM_R]

$\vdash \forall E'\ E.\ E$ ~~ $E' \Rightarrow \forall E''.\ E + E''$ ~~ $E' + E''$

[STRONG_EQUIV_SYM]

$\vdash \forall E\ E'.\ E$ ~~ $E' \Rightarrow E'$ ~~ E

[STRONG_EQUIV_TRANS]

$\vdash \forall E\ E'\ E''.\ E$ ~~ $E' \wedge E'$ ~~ $E'' \Rightarrow E$ ~~ E''

[UNION_STRONG_BISIM]

```
⊢ ∀Bsm₁ Bsm₂.
    STRONG_BISIM Bsm₁ ∧ STRONG_BISIM Bsm₂ ⇒
    STRONG_BISIM (λx y. Bsm₁ x y ∨ Bsm₂ x y)
```

3 StrongLaws Theory

Built: 14 Maggio 2017

Parent Theories: StrongEQ

3.1 Definitions

[ALL_SYNC_def]

```
⊢ (∀f f' m.
     ALL_SYNC f 0 f' m =
     SYNC (PREF_ACT (f 0)) (PREF_PROC (f 0)) f' m) ∧
  ∀f n f' m.
    ALL_SYNC f (SUC n) f' m =
    ALL_SYNC f n f' m +
    SYNC (PREF_ACT (f (SUC n))) (PREF_PROC (f (SUC n))) f' m
```

[CCS_COMP_def]

```
⊢ (∀f. PI f 0 = f 0) ∧ ∀f n. PI f (SUC n) = PI f n || f (SUC n)
```

[CCS_SIGMA_def]

$\vdash$ ($\forall f$. SIGMA f 0 = f 0) $\wedge$
$\forall f\ n$. SIGMA f (SUC n) = SIGMA f n + f (SUC n)

[Is_Prefix_def]

$\vdash \forall E$. Is_Prefix $E \iff \exists u\ E'$. E = u..E'

[PREF_ACT_def]

$\vdash \forall u\ E$. PREF_ACT (u..E) = u

[PREF_PROC_def]

$\vdash \forall u\ E$. PREF_PROC (u..E) = E

[SYNC_def]

```
⊢ (∀u P f.
     SYNC u P f 0 =
     if (u = tau) ∨ (PREF_ACT (f 0) = tau) then nil
     else if LABEL u = COMPL (LABEL (PREF_ACT (f 0))) then
       tau..(P || PREF_PROC (f 0))
     else nil) ∧
  ∀u P f n.
    SYNC u P f (SUC n) =
    if (u = tau) ∨ (PREF_ACT (f (SUC n)) = tau) then
      SYNC u P f n
    else if LABEL u = COMPL (LABEL (PREF_ACT (f (SUC n)))) then
      tau..(P || PREF_PROC (f (SUC n))) + SYNC u P f n
    else SYNC u P f n
```

3.2 Theorems

[ALL_SYNC_BASE]

```
⊢ ∀f f′ m.
    ALL_SYNC f 0 f′ m =
    SYNC (PREF_ACT (f 0)) (PREF_PROC (f 0)) f′ m
```

[ALL_SYNC_def_compute]

```
⊢ (∀f f′ m.
     ALL_SYNC f 0 f′ m =
     SYNC (PREF_ACT (f 0)) (PREF_PROC (f 0)) f′ m) ∧
  (∀f n f′ m.
     ALL_SYNC f (NUMERAL (BIT1 n)) f′ m =
     ALL_SYNC f (NUMERAL (BIT1 n) − 1) f′ m +
     SYNC (PREF_ACT (f (NUMERAL (BIT1 n))))
       (PREF_PROC (f (NUMERAL (BIT1 n)))) f′ m) ∧
  ∀f n f′ m.
    ALL_SYNC f (NUMERAL (BIT2 n)) f′ m =
    ALL_SYNC f (NUMERAL (BIT1 n)) f′ m +
    SYNC (PREF_ACT (f (NUMERAL (BIT2 n))))
      (PREF_PROC (f (NUMERAL (BIT2 n)))) f′ m
```

[ALL_SYNC_INDUCT]

$\vdash \forall f\ n\ f'\ m.$
  ALL_SYNC f (SUC n) f' m =
  ALL_SYNC f n f' m +
  SYNC (PREF_ACT (f (SUC n))) (PREF_PROC (f (SUC n))) f' m

[ALL_SYNC_TRANS_THM]

$\vdash \forall n\ m\ f\ f'\ u\ E.$
  ALL_SYNC f n f' m --u-> E $\Rightarrow$
  $\exists k\ k'\ l.$
    $k \le n \land k' \le m \land$ (PREF_ACT (f k) = label l) $\land$
    (PREF_ACT (f' k') = label (COMPL l)) $\land$ (u = tau) $\land$
    (E = PREF_PROC (f k) || PREF_PROC (f' k'))

[ALL_SYNC_TRANS_THM_EQ]

$\vdash \forall n\ m\ f\ f'\ u\ E.$
  ALL_SYNC f n f' m --u-> E $\iff$
  $\exists k\ k'\ l.$
    $k \le n \land k' \le m \land$ (PREF_ACT (f k) = label l) $\land$
    (PREF_ACT (f' k') = label (COMPL l)) $\land$ (u = tau) $\land$
    (E = PREF_PROC (f k) || PREF_PROC (f' k'))

[CCS_COMP_def_compute]

$\vdash$ ($\forall f.$ PI f 0 = f 0) $\land$
  ($\forall f\ n.$
    PI f (NUMERAL (BIT1 n)) =
    PI f (NUMERAL (BIT1 n) - 1) || f (NUMERAL (BIT1 n))) $\land$
  $\forall f\ n.$
    PI f (NUMERAL (BIT2 n)) =
    PI f (NUMERAL (BIT1 n)) || f (NUMERAL (BIT2 n))

[CCS_SIGMA_def_compute]

$\vdash$ ($\forall f.$ SIGMA f 0 = f 0) $\land$
  ($\forall f\ n.$
    SIGMA f (NUMERAL (BIT1 n)) =
    SIGMA f (NUMERAL (BIT1 n) - 1) + f (NUMERAL (BIT1 n))) $\land$
  $\forall f\ n.$
    SIGMA f (NUMERAL (BIT2 n)) =
    SIGMA f (NUMERAL (BIT1 n)) + f (NUMERAL (BIT2 n))

[COMP_BASE]

$\vdash \forall f.$ PI f 0 = f 0

[COMP_INDUCT]

$\vdash \forall f\ n.$ PI f (SUC n) = PI f n || f (SUC n)

[LESS_EQ_LESS_EQ_SUC]

$\vdash \forall m\ n.\ m \leq n \Rightarrow m \leq \texttt{SUC}\ n$

[LESS_EQ_ZERO_EQ]

$\vdash \forall n.\ n \leq 0 \Rightarrow (n = 0)$

[LESS_SUC_LESS_EQ]

$\vdash \forall m\ n.\ m < \texttt{SUC}\ n \Rightarrow m \leq n$

[LESS_SUC_LESS_EQ']

$\vdash \forall m\ n.\ m < \texttt{SUC}\ n \iff m \leq n$

[PREF_IS_PREFIX]

$\vdash \forall u\ E.\ \texttt{Is\_Prefix}\ (u..E)$

[SIGMA_BASE]

$\vdash \forall f.\ \texttt{SIGMA}\ f\ 0 = f\ 0$

[SIGMA_INDUCT]

$\vdash \forall f\ n.\ \texttt{SIGMA}\ f\ (\texttt{SUC}\ n) = \texttt{SIGMA}\ f\ n + f\ (\texttt{SUC}\ n)$

[SIGMA_TRANS_THM]

$\vdash \forall n\ f\ u\ E.\ \texttt{SIGMA}\ f\ n\ \texttt{--}u\texttt{->}\ E \Rightarrow \exists k.\ k \leq n \land f\ k\ \texttt{--}u\texttt{->}\ E$

[SIGMA_TRANS_THM_EQ]

$\vdash \forall n\ f\ u\ E.\ \texttt{SIGMA}\ f\ n\ \texttt{--}u\texttt{->}\ E \iff \exists k.\ k \leq n \land f\ k\ \texttt{--}u\texttt{->}\ E$

[STRONG_LEFT_SUM_MID_IDEMP]

$\vdash \forall E\ E'\ E''.\ E + E' + E'' + E'\ \texttt{\~{}\~{}}\ E + E'' + E'$

[STRONG_PAR_ASSOC]

$\vdash \forall E\ E'\ E''.\ E\ \texttt{||}\ E'\ \texttt{||}\ E''\ \texttt{\~{}\~{}}\ E\ \texttt{||}\ (E'\ \texttt{||}\ E'')$

[STRONG_PAR_COMM]

$\vdash \forall E\ E'.\ E\ \texttt{||}\ E'\ \texttt{\~{}\~{}}\ E'\ \texttt{||}\ E$

[STRONG_PAR_IDENT_L]

$\vdash \forall E.\ \texttt{nil}\ \texttt{||}\ E\ \texttt{\~{}\~{}}\ E$

[STRONG_PAR_IDENT_R]

$\vdash \forall E.\ E\ \texttt{||}\ \texttt{nil}\ \texttt{\~{}\~{}}\ E$

[STRONG_PAR_LAW]

$\vdash \forall f\ n\ f'\ m.$
$\quad (\forall i.\ i \leq n \Rightarrow$ `Is_Prefix` $(f\ i)) \land$
$\quad (\forall j.\ j \leq m \Rightarrow$ `Is_Prefix` $(f'\ j)) \Rightarrow$
$\quad$ `SIGMA` $f\ n$ `||` `SIGMA` $f'\ m$ `~~`
$\quad$ `SIGMA` $(\lambda i.$ `PREF_ACT` $(f\ i)$`..`(`PREF_PROC` $(f\ i)$ `||` `SIGMA` $f'\ m))$
$\qquad n$ `+`
$\quad$ `SIGMA`
$\qquad (\lambda j.$ `PREF_ACT` $(f'\ j)$`..`(`SIGMA` $f\ n$ `||` `PREF_PROC` $(f'\ j)))$
$\qquad m$ `+` `ALL_SYNC` $f\ n\ f'\ m$

[STRONG_PAR_PREF_NO_SYNCR]

$\vdash \forall l\ l'.$
$\quad l \neq$ `COMPL` $l' \Rightarrow$
$\quad \forall E\ E'.$
$\qquad$ `label` l`..`E `||` `label` l'`..`E' `~~`
$\qquad$ `label` l`..`$(E$ `||` `label` l'`..`$E')$ `+`
$\qquad$ `label` l'`..`(`label` l`..`E `||` $E')$

[STRONG_PAR_PREF_SYNCR]

$\vdash \forall l\ l'.$
$\quad (l$ `=` `COMPL` $l') \Rightarrow$
$\quad \forall E\ E'.$
$\qquad$ `label` l`..`E `||` `label` l'`..`E' `~~`
$\qquad$ `label` l`..`$(E$ `||` `label` l'`..`$E')$ `+`
$\qquad$ `label` l'`..`(`label` l`..`E `||` $E')$ `+` `tau..`$(E$ `||` $E')$

[STRONG_PAR_PREF_TAU]

$\vdash \forall u\ E\ E'.$
$\quad u$`..`E `||` `tau..`E' `~~` u`..`$(E$ `||` `tau..`$E')$ `+` `tau..`$(u$`..`E `||` $E')$

[STRONG_PAR_TAU_PREF]

$\vdash \forall E\ u\ E'.$
$\quad$ `tau..`E `||` u`..`E' `~~` `tau..`$(E$ `||` u`..`$E')$ `+` u`..`(`tau..`E `||` $E')$

[STRONG_PAR_TAU_TAU]

$\vdash \forall E\ E'.$
$\quad$ `tau..`E `||` `tau..`E' `~~`
$\quad$ `tau..`$(E$ `||` `tau..`$E')$ `+` `tau..`(`tau..`E `||` $E')$

[STRONG_PREF_REC_EQUIV]

$\vdash \forall u\ s\ v.\ u$`..rec` s $(v$`..`u`..var` $s)$ `~~` `rec` s $(u$`..`v`..var` $s)$

[STRONG_REC_ACT2]

$\vdash \forall s\ u.$ `rec` s $(u$`..`u`..var` $s)$ `~~` `rec` s $(u$`..var` $s)$

[STRONG_RELAB_NIL]

$\vdash \forall rf.$ relab nil rf ~~ nil

[STRONG_RELAB_PREFIX]

$\vdash \forall u\ E\ labl.$
relab (u..E) (RELAB $labl$) ~~
relabel (RELAB $labl$) u..relab E (RELAB $labl$)

[STRONG_RELAB_SUM]

$\vdash \forall E\ E'\ rf.$ relab (E + E') rf ~~ relab E rf + relab E' rf

[STRONG_RESTR_NIL]

$\vdash \forall L.$ nu L nil ~~ nil

[STRONG_RESTR_PR_LAB_NIL]

$\vdash \forall l\ L.\ l \in L \lor$ COMPL $l \in L \Rightarrow \forall E.$ nu L (label l..E) ~~ nil

[STRONG_RESTR_PREFIX_LABEL]

$\vdash \forall l\ L.$
$l \notin L \land$ COMPL $l \notin L \Rightarrow$
$\forall E.$ nu L (label l..E) ~~ label l..nu L E

[STRONG_RESTR_PREFIX_TAU]

$\vdash \forall E\ L.$ nu L (tau..E) ~~ tau..nu L E

[STRONG_RESTR_SUM]

$\vdash \forall E\ E'\ L.$ nu L (E + E') ~~ nu L E + nu L E'

[STRONG_SUM_ASSOC_L]

$\vdash \forall E\ E'\ E''.\ E$ + (E' + E'') ~~ E + E' + E''

[STRONG_SUM_ASSOC_R]

$\vdash \forall E\ E'\ E''.\ E$ + E' + E'' ~~ E + (E' + E'')

[STRONG_SUM_COMM]

$\vdash \forall E\ E'.\ E$ + E' ~~ E' + E

[STRONG_SUM_IDEMP]

$\vdash \forall E.\ E$ + E ~~ E

[STRONG_SUM_IDENT_L]

$\vdash \forall E.$ nil + E ~~ E

[STRONG_SUM_IDENT_R]

$\vdash \forall E.\ E$ + nil ~~ E

[STRONG_SUM_MID_IDEMP]

```
⊢ ∀E E'. E + E' + E ~~ E' + E
```

[STRONG_UNFOLDING]

```
⊢ ∀X E. rec X E ~~ CCS_Subst E (rec X E) X
```

[SYNC_BASE]

```
⊢ ∀u P f.
    SYNC u P f 0 =
    if (u = tau) ∨ (PREF_ACT (f 0) = tau) then nil
    else if LABEL u = COMPL (LABEL (PREF_ACT (f 0))) then
      tau..(P || PREF_PROC (f 0))
    else nil
```

[SYNC_def_compute]

```
⊢ (∀u P f.
     SYNC u P f 0 =
     if (u = tau) ∨ (PREF_ACT (f 0) = tau) then nil
     else if LABEL u = COMPL (LABEL (PREF_ACT (f 0))) then
       tau..(P || PREF_PROC (f 0))
     else nil) ∧
  (∀u P f n.
     SYNC u P f (NUMERAL (BIT1 n)) =
     if
       (u = tau) ∨ (PREF_ACT (f (NUMERAL (BIT1 n))) = tau)
     then
       SYNC u P f (NUMERAL (BIT1 n) - 1)
     else if
       LABEL u =
       COMPL (LABEL (PREF_ACT (f (NUMERAL (BIT1 n)))))
     then
       tau..(P || PREF_PROC (f (NUMERAL (BIT1 n)))) +
       SYNC u P f (NUMERAL (BIT1 n) - 1)
     else SYNC u P f (NUMERAL (BIT1 n) - 1)) ∧
  ∀u P f n.
    SYNC u P f (NUMERAL (BIT2 n)) =
    if (u = tau) ∨ (PREF_ACT (f (NUMERAL (BIT2 n))) = tau) then
      SYNC u P f (NUMERAL (BIT1 n))
    else if
      LABEL u = COMPL (LABEL (PREF_ACT (f (NUMERAL (BIT2 n)))))
    then
      tau..(P || PREF_PROC (f (NUMERAL (BIT2 n)))) +
      SYNC u P f (NUMERAL (BIT1 n))
    else SYNC u P f (NUMERAL (BIT1 n))
```

[SYNC_INDUCT]

```
⊢ ∀u P f n.
    SYNC u P f (SUC n) =
    if (u = tau) ∨ (PREF_ACT (f (SUC n)) = tau) then
      SYNC u P f n
    else if LABEL u = COMPL (LABEL (PREF_ACT (f (SUC n)))) then
      tau..(P || PREF_PROC (f (SUC n))) + SYNC u P f n
    else SYNC u P f n
```

[SYNC_TRANS_THM]

```
⊢ ∀m u P f v Q.
    SYNC u P f m --v-> Q ⇒
    ∃j l.
      j ≤ m ∧ (u = label l) ∧
      (PREF_ACT (f j) = label (COMPL l)) ∧ (v = tau) ∧
      (Q = P || PREF_PROC (f j))
```

[SYNC_TRANS_THM_EQ]

```
⊢ ∀m u P f v Q.
    SYNC u P f m --v-> Q  ⟺
    ∃j l.
      j ≤ m ∧ (u = label l) ∧
      (PREF_ACT (f j) = label (COMPL l)) ∧ (v = tau) ∧
      (Q = P || PREF_PROC (f j))
```

4 WeakEQ Theory

Built: 14 Maggio 2017
Parent Theories: StrongEQ

4.1 Definitions

[EPS1_def]

```
⊢ EPS1 =
  (λa0 a1.
     ∀EPS1'.
       (∀a0 a1. a0 --tau-> a1 ⇒ EPS1' a0 a1) ⇒ EPS1' a0 a1)
```

[EPS_def]

```
⊢ EPS = EPS1*
```

[epsilon_def]

```
⊢ epsilon = []
```

[RWEAK_EQUIV]

```
⊢ ∀E E'.
    E ~~c E' ⇔
    ∀u.
      (∀E1. E --u-> E1 ⇒ ∃E2. E' ==u=>> E2 ∧ E1 ~~~ E2) ∧
      ∀E2. E' --u-> E2 ⇒ ∃E1. E ==u=>> E1 ∧ E1 ~~~ E2
```

[TRACE_def]

```
⊢ TRACE =
  (λa0 a1 a2.
     ∀TRACE'.
       (∀a0 a1 a2.
          (a1 = epsilon) ∧ (a2 = a0) ∨
          (∃l. (a1 = [l]) ∧ a0 --label l-> a2) ∨
          (∃E2 l1 l2.
             (a1 = l1 ++ l2) ∧ TRACE' a0 l1 E2 ∧
             TRACE' E2 l2 a2) ⇒
          TRACE' a0 a1 a2) ⇒
       TRACE' a0 a1 a2)
```

[WEAK_BISIM]

```
⊢ ∀Wbsm.
    WEAK_BISIM Wbsm ⇔
    ∀E E'.
      Wbsm E E' ⇒
      (∀l.
         (∀E1.
            E --label l-> E1 ⇒
            ∃E2. E' ==label l=>> E2 ∧ Wbsm E1 E2) ∧
         ∀E2.
           E' --label l-> E2 ⇒
           ∃E1. E ==label l=>> E1 ∧ Wbsm E1 E2) ∧
      (∀E1. E --tau-> E1 ⇒ ∃E2. EPS E' E2 ∧ Wbsm E1 E2) ∧
      ∀E2. E' --tau-> E2 ⇒ ∃E1. EPS E E1 ∧ Wbsm E1 E2
```

[WEAK_EQUIV_def]

```
⊢ WEAK_EQUIV =
  (λa0 a1.
     ∃WEAK_EQUIV'.
       WEAK_EQUIV' a0 a1 ∧
       ∀a0 a1.
         WEAK_EQUIV' a0 a1 ⇒
         (∀l.
            (∀E1.
               a0 --label l-> E1 ⇒
               ∃E2. a1 ==label l=>> E2 ∧ WEAK_EQUIV' E1 E2) ∧
            ∀E2.
              a1 --label l-> E2 ⇒
```

$\exists E_1$. a_0 ==label l=>> E_1 ∧ $WEAK\_EQUIV'$ E_1 E_2) ∧
(∀ E_1.
a_0 --tau-> E_1 ⇒
∃ E_2. EPS a_1 E_2 ∧ $WEAK\_EQUIV'$ E_1 E_2) ∧
∀ E_2.
a_1 --tau-> E_2 ⇒ ∃ E_1. EPS a_0 E_1 ∧ $WEAK\_EQUIV'$ E_1 E_2)

[WEAK_TRACE_def]

```
⊢ WEAK_TRACE =
  (λ a0 a1 a2.
     ∀ WEAK_TRACE'.
       (∀ a0 a1 a2.
          (a1 = epsilon) ∧ (a2 = a0) ∨
          (a1 = epsilon) ∧ a0 --tau-> a2 ∨
          (∃ l. (a1 = [l]) ∧ a0 --label l-> a2) ∨
          (∃ E2 l1 l2.
             (a1 = l1 ++ l2) ∧ WEAK_TRACE' a0 l1 E2 ∧
             WEAK_TRACE' E2 l2 a2) ⇒
          WEAK_TRACE' a0 a1 a2) ⇒
       WEAK_TRACE' a0 a1 a2)
```

[WEAK_TRANS_def]

```
⊢ WEAK_TRANS =
  (λ a0 a1 a2.
     ∀ WEAK_TRANS'.
       (∀ a0 a1 a2.
          (∃ E1 E2. EPS a0 E1 ∧ E1 --a1-> E2 ∧ EPS E2 a2) ⇒
          WEAK_TRANS' a0 a1 a2) ⇒
       WEAK_TRANS' a0 a1 a2)
```

4.2 Theorems

[EPS1_cases]

⊢ ∀ a_0 a_1. EPS1 a_0 a_1 ⟺ a_0 --tau-> a_1

[EPS1_ind]

```
⊢ ∀ EPS1'.
    (∀ E E'. E --tau-> E' ⇒ EPS1' E E') ⇒
    ∀ a0 a1. EPS1 a0 a1 ⇒ EPS1' a0 a1
```

[EPS1_rules]

⊢ ∀ E E'. E --tau-> E' ⇒ EPS1 E E'

[EPS1_strongind]

```
⊢ ∀ EPS1'.
    (∀ E E'. E --tau-> E' ⇒ EPS1' E E') ⇒
    ∀ a0 a1. EPS1 a0 a1 ⇒ EPS1' a0 a1
```

[TRACE_cases]

$\vdash \forall a_0\ a_1\ a_2.$
$\quad \texttt{TRACE}\ a_0\ a_1\ a_2 \iff$
$\quad (a_1 = \texttt{epsilon}) \land (a_2 = a_0) \lor$
$\quad (\exists l.\ (a_1 = [l]) \land a_0\ \texttt{--label}\ l\texttt{->}\ a_2) \lor$
$\quad \exists E_2\ l_1\ l_2.$
$\quad\quad (a_1 = l_1\ \texttt{++}\ l_2) \land \texttt{TRACE}\ a_0\ l_1\ E_2 \land \texttt{TRACE}\ E_2\ l_2\ a_2$

[TRACE_ind]

$\vdash \forall \mathit{TRACE}'.$
$\quad (\forall E.\ \mathit{TRACE}'\ E\ \texttt{epsilon}\ E) \land$
$\quad (\forall E\ E'\ l.\ E\ \texttt{--label}\ l\texttt{->}\ E' \Rightarrow \mathit{TRACE}'\ E\ [l]\ E') \land$
$\quad (\forall E_1\ E_2\ E_3\ l_1\ l_2.$
$\quad\quad \mathit{TRACE}'\ E_1\ l_1\ E_2 \land \mathit{TRACE}'\ E_2\ l_2\ E_3 \Rightarrow$
$\quad\quad \mathit{TRACE}'\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3) \Rightarrow$
$\quad \forall a_0\ a_1\ a_2.\ \texttt{TRACE}\ a_0\ a_1\ a_2 \Rightarrow \mathit{TRACE}'\ a_0\ a_1\ a_2$

[TRACE_rules]

$\vdash (\forall E.\ \texttt{TRACE}\ E\ \texttt{epsilon}\ E) \land$
$\quad (\forall E\ E'\ l.\ E\ \texttt{--label}\ l\texttt{->}\ E' \Rightarrow \texttt{TRACE}\ E\ [l]\ E') \land$
$\quad \forall E_1\ E_2\ E_3\ l_1\ l_2.$
$\quad\quad \texttt{TRACE}\ E_1\ l_1\ E_2 \land \texttt{TRACE}\ E_2\ l_2\ E_3 \Rightarrow \texttt{TRACE}\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3$

[TRACE_strongind]

$\vdash \forall \mathit{TRACE}'.$
$\quad (\forall E.\ \mathit{TRACE}'\ E\ \texttt{epsilon}\ E) \land$
$\quad (\forall E\ E'\ l.\ E\ \texttt{--label}\ l\texttt{->}\ E' \Rightarrow \mathit{TRACE}'\ E\ [l]\ E') \land$
$\quad (\forall E_1\ E_2\ E_3\ l_1\ l_2.$
$\quad\quad \texttt{TRACE}\ E_1\ l_1\ E_2 \land \mathit{TRACE}'\ E_1\ l_1\ E_2 \land \texttt{TRACE}\ E_2\ l_2\ E_3 \land$
$\quad\quad \mathit{TRACE}'\ E_2\ l_2\ E_3 \Rightarrow$
$\quad\quad \mathit{TRACE}'\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3) \Rightarrow$
$\quad \forall a_0\ a_1\ a_2.\ \texttt{TRACE}\ a_0\ a_1\ a_2 \Rightarrow \mathit{TRACE}'\ a_0\ a_1\ a_2$

[WEAK_EQUIV]

$\vdash \forall E\ E'.\ E\ \texttt{\~{}\~{}\~{}}\ E' \iff \exists \mathit{Wbsm}.\ \mathit{Wbsm}\ E\ E' \land \texttt{WEAK\_BISIM}\ \mathit{Wbsm}$

[WEAK_EQUIV_cases]

$\vdash \forall a_0\ a_1.$
$\quad a_0\ \texttt{\~{}\~{}\~{}}\ a_1 \iff$
$\quad (\forall l.$
$\quad\quad (\forall E_1.$
$\quad\quad\quad a_0\ \texttt{--label}\ l\texttt{->}\ E_1 \Rightarrow$
$\quad\quad\quad \exists E_2.\ a_1\ \texttt{==label}\ l\texttt{=>>}\ E_2 \land E_1\ \texttt{\~{}\~{}\~{}}\ E_2) \land$
$\quad\quad \forall E_2.$
$\quad\quad\quad a_1\ \texttt{--label}\ l\texttt{->}\ E_2 \Rightarrow$
$\quad\quad\quad \exists E_1.\ a_0\ \texttt{==label}\ l\texttt{=>>}\ E_1 \land E_1\ \texttt{\~{}\~{}\~{}}\ E_2) \land$
$\quad (\forall E_1.\ a_0\ \texttt{--tau->}\ E_1 \Rightarrow \exists E_2.\ \texttt{EPS}\ a_1\ E_2 \land E_1\ \texttt{\~{}\~{}\~{}}\ E_2) \land$
$\quad \forall E_2.\ a_1\ \texttt{--tau->}\ E_2 \Rightarrow \exists E_1.\ \texttt{EPS}\ a_0\ E_1 \land E_1\ \texttt{\~{}\~{}\~{}}\ E_2$

[WEAK_EQUIV_coind]

$\vdash \forall \mathit{WEAK\_EQUIV}'.$
$(\forall a_0\ a_1.$
$\mathit{WEAK\_EQUIV}'\ a_0\ a_1 \Rightarrow$
$(\forall l.$
$(\forall E_1.$
a_0 `--label` l`->` $E_1 \Rightarrow$
$\exists E_2.\ a_1$ `==label` l`=>>` $E_2 \land \mathit{WEAK\_EQUIV}'\ E_1\ E_2) \land$
$\forall E_2.$
a_1 `--label` l`->` $E_2 \Rightarrow$
$\exists E_1.\ a_0$ `==label` l`=>>` $E_1 \land \mathit{WEAK\_EQUIV}'\ E_1\ E_2) \land$
$(\forall E_1.$
a_0 `--tau->` $E_1 \Rightarrow \exists E_2.$ `EPS` $a_1\ E_2 \land \mathit{WEAK\_EQUIV}'\ E_1\ E_2) \land$
$\forall E_2.$
a_1 `--tau->` $E_2 \Rightarrow \exists E_1.$ `EPS` $a_0\ E_1 \land \mathit{WEAK\_EQUIV}'\ E_1\ E_2) \Rightarrow$
$\forall a_0\ a_1.\ \mathit{WEAK\_EQUIV}'\ a_0\ a_1 \Rightarrow a_0$ `~~~` a_1

[WEAK_EQUIV_IS_WEAK_BISIM]

$\vdash$ `WEAK_BISIM WEAK_EQUIV`

[WEAK_EQUIV_rules]

$\vdash \forall E\ E'.$
$(\forall l.$
$(\forall E_1.$
E `--label` l`->` $E_1 \Rightarrow$
$\exists E_2.\ E'$ `==label` l`=>>` $E_2 \land E_1$ `~~~` $E_2) \land$
$\forall E_2.$
E' `--label` l`->` $E_2 \Rightarrow$
$\exists E_1.\ E$ `==label` l`=>>` $E_1 \land E_1$ `~~~` $E_2) \land$
$(\forall E_1.\ E$ `--tau->` $E_1 \Rightarrow \exists E_2.$ `EPS` $E'\ E_2 \land E_1$ `~~~` $E_2) \land$
$(\forall E_2.\ E'$ `--tau->` $E_2 \Rightarrow \exists E_1.$ `EPS` $E\ E_1 \land E_1$ `~~~` $E_2) \Rightarrow$
E `~~~` E'

[WEAK_TRACE_cases]

$\vdash \forall a_0\ a_1\ a_2.$
`WEAK_TRACE` $a_0\ a_1\ a_2 \iff$
$(a_1$ `= epsilon`$) \land (a_2 = a_0) \lor$
$(a_1$ `= epsilon`$) \land a_0$ `--tau->` $a_2 \lor$
$(\exists l.\ (a_1 = [l]) \land a_0$ `--label` l`->` $a_2) \lor$
$\exists E_2\ l_1\ l_2.$
$(a_1 = l_1$ `++` $l_2) \land$ `WEAK_TRACE` $a_0\ l_1\ E_2 \land$
`WEAK_TRACE` $E_2\ l_2\ a_2$

[WEAK_TRACE_ind]

$\vdash \forall \mathit{WEAK\_TRACE}'.$
$(\forall E.\ \mathit{WEAK\_TRACE}'\ E$ `epsilon` $E) \land$
$(\forall E\ E'.\ E$ `--tau->` $E' \Rightarrow \mathit{WEAK\_TRACE}'\ E$ `epsilon` $E') \land$
$(\forall E\ E'\ l.\ E$ `--label` l`->` $E' \Rightarrow \mathit{WEAK\_TRACE}'\ E\ [l]\ E') \land$

$$(\forall E_1\ E_2\ E_3\ l_1\ l_2.\ \mathit{WEAK\_TRACE}'\ E_1\ l_1\ E_2 \land \mathit{WEAK\_TRACE}'\ E_2\ l_2\ E_3 \Rightarrow \mathit{WEAK\_TRACE}'\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3) \Rightarrow$$
$$\forall a_0\ a_1\ a_2.\ \texttt{WEAK\_TRACE}\ a_0\ a_1\ a_2 \Rightarrow \mathit{WEAK\_TRACE}'\ a_0\ a_1\ a_2$$

[WEAK_TRACE_rules]

$$\vdash (\forall E.\ \texttt{WEAK\_TRACE}\ E\ \texttt{epsilon}\ E) \land (\forall E\ E'.\ E\ \texttt{--tau->}\ E' \Rightarrow \texttt{WEAK\_TRACE}\ E\ \texttt{epsilon}\ E') \land (\forall E\ E'\ l.\ E\ \texttt{--label}\ l\texttt{->}\ E' \Rightarrow \texttt{WEAK\_TRACE}\ E\ [l]\ E') \land \forall E_1\ E_2\ E_3\ l_1\ l_2.\ \texttt{WEAK\_TRACE}\ E_1\ l_1\ E_2 \land \texttt{WEAK\_TRACE}\ E_2\ l_2\ E_3 \Rightarrow \texttt{WEAK\_TRACE}\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3$$

[WEAK_TRACE_strongind]

$$\vdash \forall \mathit{WEAK\_TRACE}'.\ (\forall E.\ \mathit{WEAK\_TRACE}'\ E\ \texttt{epsilon}\ E) \land (\forall E\ E'.\ E\ \texttt{--tau->}\ E' \Rightarrow \mathit{WEAK\_TRACE}'\ E\ \texttt{epsilon}\ E') \land (\forall E\ E'\ l.\ E\ \texttt{--label}\ l\texttt{->}\ E' \Rightarrow \mathit{WEAK\_TRACE}'\ E\ [l]\ E') \land (\forall E_1\ E_2\ E_3\ l_1\ l_2.\ \texttt{WEAK\_TRACE}\ E_1\ l_1\ E_2 \land \mathit{WEAK\_TRACE}'\ E_1\ l_1\ E_2 \land \texttt{WEAK\_TRACE}\ E_2\ l_2\ E_3 \land \mathit{WEAK\_TRACE}'\ E_2\ l_2\ E_3 \Rightarrow \mathit{WEAK\_TRACE}'\ E_1\ (l_1\ \texttt{++}\ l_2)\ E_3) \Rightarrow \forall a_0\ a_1\ a_2.\ \texttt{WEAK\_TRACE}\ a_0\ a_1\ a_2 \Rightarrow \mathit{WEAK\_TRACE}'\ a_0\ a_1\ a_2$$

[WEAK_TRANS_cases]

$$\vdash \forall a_0\ a_1\ a_2.\ a_0\ \texttt{==}a_1\texttt{=>>}\ a_2 \iff \exists E_1\ E_2.\ \texttt{EPS}\ a_0\ E_1 \land E_1\ \texttt{--}a_1\texttt{->}\ E_2 \land \texttt{EPS}\ E_2\ a_2$$

[WEAK_TRANS_ind]

$$\vdash \forall \mathit{WEAK\_TRANS}'.\ (\forall E\ E'\ E_1\ E_2\ u.\ \texttt{EPS}\ E\ E_1 \land E_1\ \texttt{--}u\texttt{->}\ E_2 \land \texttt{EPS}\ E_2\ E' \Rightarrow \mathit{WEAK\_TRANS}'\ E\ u\ E') \Rightarrow \forall a_0\ a_1\ a_2.\ a_0\ \texttt{==}a_1\texttt{=>>}\ a_2 \Rightarrow \mathit{WEAK\_TRANS}'\ a_0\ a_1\ a_2$$

[WEAK_TRANS_rules]

$$\vdash \forall E\ E'\ E_1\ E_2\ u.\ \texttt{EPS}\ E\ E_1 \land E_1\ \texttt{--}u\texttt{->}\ E_2 \land \texttt{EPS}\ E_2\ E' \Rightarrow E\ \texttt{==}u\texttt{=>>}\ E'$$

[WEAK_TRANS_strongind]

$$\vdash \forall \mathit{WEAK\_TRANS}'.\ (\forall E\ E'\ E_1\ E_2\ u.\ \texttt{EPS}\ E\ E_1 \land E_1\ \texttt{--}u\texttt{->}\ E_2 \land \texttt{EPS}\ E_2\ E' \Rightarrow \mathit{WEAK\_TRANS}'\ E\ u\ E') \Rightarrow \forall a_0\ a_1\ a_2.\ a_0\ \texttt{==}a_1\texttt{=>>}\ a_2 \Rightarrow \mathit{WEAK\_TRANS}'\ a_0\ a_1\ a_2$$

5 Example Theory

Built: 14 Maggio 2017
Parent Theories: WeakEQ, StrongLaws

5.1 Theorems

[ex1]

```
⊢ label (name “a”)..label (name “b”)..nil +
  label (name “b”)..label (name “a”)..nil
  --label (name “a”)->
  label (name “b”)..nil
```

[ex1’]

```
⊢ label (name “a”)..label (name “b”)..nil +
  label (name “b”)..label (name “a”)..nil
  --label (name “a”)->
  label (name “b”)..nil
```

[ex1’’]

```
⊢ In “a”..In “b”..nil + In “b”..In “a”..nil
  --In “a”->
  In “b”..nil
```

[ex2]

```
⊢ label (name “a”)..label (name “b”)..nil +
  label (name “b”)..label (name “a”)..nil
  --label (name “b”)->
  label (name “a”)..nil
```

[ex3]

```
⊢ nu {name “c”}
    (label (name “a”)..label (name “c”)..nil ||
     (label (coname “a”)..nil + label (name “c”)..nil))
  --tau->
  nu {name “c”} (label (name “c”)..nil || nil)
```

[ex4]

```
⊢ nu {name “c”}
    (label (name “a”)..label (name “c”)..nil ||
     (label (name “b”)..nil + label (name “c”)..nil))
  --label (name “b”)->
  nu {name “c”}
    (label (name “a”)..label (name “c”)..nil || nil)
```

[ex5]

```
⊢ label (name “a”)..nil || label (coname “a”)..nil || nil
  --tau->
  nil || nil || nil
```

[ex6]

```
⊢ ¬(nu {name “c”}
       (label (name “a”)..label (name “c”)..nil ||
        label (name “b”)..nil)
    --tau->
    nu {name “c”} (label (name “c”)..nil || nil))
```

[ex7]

```
⊢ ¬(nu {name “a”}
       (label (name “a”)..nil || label (coname “a”)..nil)
    --label (name “a”)->
    nu {name “a”} (nil || label (coname “a”)..nil))
```

[ex_A]

```
⊢ ∀u E.
    label (name “a”)..nil || label (coname “a”)..nil --u-> E ⟺
    ((u = label (name “a”)) ∧
     (E = nil || label (coname “a”)..nil) ∨
     (u = label (coname “a”)) ∧
     (E = label (name “a”)..nil || nil)) ∨
    (u = tau) ∧ (E = nil || nil)
```

[ex_A1]

```
⊢ ∀u E.
    nil || label (coname “a”)..nil --u-> E ⟺
    (u = label (coname “a”)) ∧ (E = nil || nil)
```

[ex_A2]

```
⊢ ∀u E.
    label (name “a”)..nil || nil --u-> E ⟺
    (u = label (name “a”)) ∧ (E = nil || nil)
```

[ex_A3]

```
⊢ ∀u E. ¬(nil || nil --u-> E)
```

[ex_B]

```
⊢ ∀u E.
    nu {name “a”}
      (label (name “a”)..nil || label (coname “a”)..nil)
    --u->
    E ⟺ (u = tau) ∧ (E = nu {name “a”} (nil || nil))
```

[ex_B0]

```
⊢ ∀u E. ¬(nu {name “a”} (nil || nil) --u-> E)
```

[ex_C]

```
⊢ ∀u E.
     nu {name “a”}
       (label (name “a”)..nil || label (coname “a”)..nil) ||
     label (name “a”)..nil
     --u->
     E ⇔
     (u = tau) ∧
     (E = nu {name “a”} (nil || nil) || label (name “a”)..nil) ∨
     (u = label (name “a”)) ∧
     (E =
      nu {name “a”}
        (label (name “a”)..nil || label (coname “a”)..nil) ||
      nil)
```

[ex_C1]

```
⊢ ∀u E.
     nu {name “a”} (nil || nil) || label (name “a”)..nil
     --u->
     E ⇔
     (u = label (name “a”)) ∧
     (E = nu {name “a”} (nil || nil) || nil)
```

[ex_C2]

```
⊢ ∀u E.
     nu {name “a”}
       (label (name “a”)..nil || label (coname “a”)..nil) || nil
     --u->
     E ⇔ (u = tau) ∧ (E = nu {name “a”} (nil || nil) || nil)
```

[par_nils_no_trans]

```
⊢ ∀l G. ¬(nil || nil --l-> G)
```

[r1_has_no_other_trans]

```
⊢ ¬∃l G.
      ¬((G = nil || label (coname “a”)..nil) ∧
        (l = label (name “a”)) ∨
        (G = label (name “a”)..nil || nil) ∧
        (l = label (coname “a”)) ∨
        (G = nil || nil) ∧ (l = tau)) ∧
      label (name “a”)..nil || label (coname “a”)..nil --l-> G
```

[r1_has_trans]

```
⊢ ∃l G.
     label (name “a”)..nil || label (coname “a”)..nil --l-> G
```

[r1_s1_has_no_other_trans]
```
⊢ ¬∃l G.
      ¬((G = nil || nil) ∧ (l = label (coname “a”))) ∧
      nil || label (coname “a”)..nil --l-> G
```

[r1_s1_has_trans]
```
⊢ ∃l G. label (name “a”)..nil || nil --l-> G
```

[r1_s2_has_no_other_trans]
```
⊢ ¬∃l G.
      ¬((G = nil || nil) ∧ (l = label (name “a”))) ∧
      label (name “a”)..nil || nil --l-> G
```

[r1_trans_1]
```
⊢ label (name “a”)..nil || label (coname “a”)..nil
  --label (name “a”)->
  nil || label (coname “a”)..nil
```

[r1_trans_2]
```
⊢ label (name “a”)..nil || label (coname “a”)..nil
  --label (coname “a”)->
  label (name “a”)..nil || nil
```

[r1_trans_3]
```
⊢ label (name “a”)..nil || label (coname “a”)..nil
  --tau->
  nil || nil
```

[r2_final_no_trans]
```
⊢ ∀l G. ¬(nu {name “a”} (nil || nil) --l-> G)
```

[r2_has_no_other_trans]
```
⊢ ¬∃l G.
      ¬((l = tau) ∧ (G = nu {name “a”} (nil || nil))) ∧
      nu {name “a”}
        (label (name “a”)..nil || label (coname “a”)..nil)
      --l->
      G
```

[r2_has_trans]
```
⊢ ∃l G.
     nu {name “a”}
       (label (name “a”)..nil || label (coname “a”)..nil)
     --l->
     G
```

[r2_trans]
```
⊢ nu {name “a”}
    (label (name “a”)..nil || label (coname “a”)..nil)
  --tau->
  nu {name “a”} (nil || nil)
```

Index

CCS Theory, 3
Datatypes, 3
Definitions, 3
Apply_Relab_def, 3
ARB'_def, 3
CCS_Subst_def, 3
COMPL_ACT_def, 4
COMPL_LAB_def, 4
In_def, 4
IS_LABEL_def, 4
Is_Relabeling_def, 4
LABEL_def, 4
Out_def, 4
RELAB_def, 4
relabel_def, 4
Relabeling_ISO_DEF, 4
Relabeling_TY_DEF, 4
Restr_def, 4
TRANS_def, 5
Theorems, 5
Action_distinct_label, 5
Action_not_tau_is_Label, 5
Apply_Relab_COMPL_THM, 5
APPLY_RELAB_THM, 5
CCS_COND_CLAUSES, 6
CCS_distinct', 6
COMPL_COMPL_ACT, 7
COMPL_COMPL_LAB, 7
COMPL_THM, 7
coname_COMPL, 7
EXISTS_Relabeling, 7
IS_RELABELING, 7
Label_distinct', 7
Label_not_eq, 7
Label_not_eq', 7
NIL_NO_TRANS, 7
NIL_NO_TRANS_EQF, 7
PAR1, 7
PAR2, 7
PAR3, 8
PAR_cases, 8
PAR_cases_EQ, 8
PREFIX, 8
REC, 8
REC_cases, 8
REC_cases_EQ, 8
RELAB, 9
RELAB_cases, 9
RELAB_cases_EQ, 9
RELAB_NIL_NO_TRANS, 9
REP_Relabeling_THM, 9
RESTR, 9
RESTR_cases, 9
RESTR_cases_EQ, 9
RESTR_LABEL_NO_TRANS, 9
RESTR_NIL_NO_TRANS, 10
SUM1, 10
SUM2, 10
SUM_cases, 10
SUM_cases_EQ, 10
TRANS_ASSOC_EQ, 10
TRANS_ASSOC_RL, 10
TRANS_cases, 10
TRANS_COMM_EQ, 11
TRANS_IMP_NO_NIL, 11
TRANS_IMP_NO_NIL', 11
TRANS_IMP_NO_RESTR_NIL, 11
TRANS_ind, 11
TRANS_P_RESTR, 11
TRANS_P_SUM_P, 11
TRANS_P_SUM_P_EQ, 11
TRANS_PAR, 12
TRANS_PAR_EQ, 12
TRANS_PAR_NO_SYNCR, 12
TRANS_PAR_P_NIL, 12
TRANS_PREFIX, 12
TRANS_PREFIX_EQ, 12
TRANS_REC, 12
TRANS_REC_EQ, 12
TRANS_RELAB, 12

TRANS_RELAB_EQ, 13
TRANS_RELAB_labl, 13
TRANS_RESTR, 13
TRANS_RESTR_EQ, 13
TRANS_RESTR_NO_NIL, 13
TRANS_rules, 13
TRANS_strongind, 14
TRANS_SUM, 14
TRANS_SUM_EQ, 14
TRANS_SUM_EQ', 14
TRANS_SUM_NIL, 14
TRANS_SUM_NIL_EQ, 14
VAR_NO_TRANS, 14

Example Theory, 32
Theorems, 32
ex1, 32
ex1', 32
ex1'', 32
ex2, 32
ex3, 32
ex4, 32
ex5, 33
ex6, 33
ex7, 33
ex_A, 33
ex_A1, 33
ex_A2, 33
ex_A3, 33
ex_B, 33
ex_B0, 34
ex_C, 34
ex_C1, 34
ex_C2, 34
par_nils_no_trans, 34
r1_has_no_other_trans, 34
r1_has_trans, 34
r1_s1_has_no_other_trans, 35
r1_s1_has_trans, 35
r1_s2_has_no_other_trans, 35
r1_trans_1, 35
r1_trans_2, 35
r1_trans_3, 35
r2_final_no_trans, 35
r2_has_no_other_trans, 35
r2_has_trans, 35
r2_trans, 35

StrongEQ Theory, 15
Definitions, 15
BIGUNION_BISIM_def, 15
STRONG_BISIM, 15
STRONG_BISIM_UPTO, 15
STRONG_EQ_def, 15
STRONG_EQUIV, 16
STRONG_EQUIV', 16
Theorems, 16
COMP_STRONG_BISIM, 16
CONVERSE_STRONG_BISIM, 16
EQUAL_IMP_STRONG_EQUIV, 16
IDENTITY_STRONG_BISIM, 16
PROPERTY_STAR, 16
PROPERTY_STAR_LR, 16
STR_EQ_IMP_STR_EQUIV, 16
STR_EQ_IS_STR_BISIM, 17
STR_EQ_TO_STR_EQUIV, 17
STR_EQUIV'_IMP_STR_EQUIV, 17
STR_EQUIV'_TO_STR_EQUIV, 17
STR_EQUIV_IMP_STR_EQ, 17
STR_EQUIV_IMP_STR_EQUIV', 17
STR_EQUIV_TO_STR_EQ, 17
STR_EQUIV_TO_STR_EQUIV', 17
STRONG_EQ_cases, 17
STRONG_EQ_coind, 17
STRONG_EQ_rules, 18
STRONG_EQUIV_coind, 18
STRONG_EQUIV_EQ_BIGUNION_BISIM, 18
STRONG_EQUIV_IS_BIGUNION_BISIM, 18
STRONG_EQUIV_IS_BIGUNION_BISIM', 18
STRONG_EQUIV_IS_BIGUNION_BISIM'', 18
STRONG_EQUIV_IS_STRONG_BISIM, 18
STRONG_EQUIV_PRESD_BY_PAR, 18

STRONG_EQUIV_PRESD_BY_SUM, 18
STRONG_EQUIV_REFL, 18
STRONG_EQUIV_SUBST_PAR_L, 18
STRONG_EQUIV_SUBST_PAR_R, 18
STRONG_EQUIV_SUBST_PREFIX, 19
STRONG_EQUIV_SUBST_RELAB, 19
STRONG_EQUIV_SUBST_RESTR, 19
STRONG_EQUIV_SUBST_SUM_L, 19
STRONG_EQUIV_SUBST_SUM_R, 19
STRONG_EQUIV_SYM, 19
STRONG_EQUIV_TRANS, 19
UNION_STRONG_BISIM, 19

StrongLaws Theory, 19
Definitions, 19
ALL_SYNC_def, 19
CCS_COMP_def, 19
CCS_SIGMA_def, 20
Is_Prefix_def, 20
PREF_ACT_def, 20
PREF_PROC_def, 20
SYNC_def, 20
Theorems, 20
ALL_SYNC_BASE, 20
ALL_SYNC_def_compute, 20
ALL_SYNC_INDUCT, 21
ALL_SYNC_TRANS_THM, 21
ALL_SYNC_TRANS_THM_EQ, 21
CCS_COMP_def_compute, 21
CCS_SIGMA_def_compute, 21
COMP_BASE, 21
COMP_INDUCT, 21
LESS_EQ_LESS_EQ_SUC, 22
LESS_EQ_ZERO_EQ, 22
LESS_SUC_LESS_EQ, 22
LESS_SUC_LESS_EQ', 22
PREF_IS_PREFIX, 22
SIGMA_BASE, 22
SIGMA_INDUCT, 22
SIGMA_TRANS_THM, 22
SIGMA_TRANS_THM_EQ, 22
STRONG_LEFT_SUM_MID_IDEMP, 22
STRONG_PAR_ASSOC, 22
STRONG_PAR_COMM, 22
STRONG_PAR_IDENT_L, 22
STRONG_PAR_IDENT_R, 22
STRONG_PAR_LAW, 23
STRONG_PAR_PREF_NO_SYNCR, 23
STRONG_PAR_PREF_SYNCR, 23
STRONG_PAR_PREF_TAU, 23
STRONG_PAR_TAU_PREF, 23
STRONG_PAR_TAU_TAU, 23
STRONG_PREF_REC_EQUIV, 23
STRONG_REC_ACT2, 23
STRONG_RELAB_NIL, 24
STRONG_RELAB_PREFIX, 24
STRONG_RELAB_SUM, 24
STRONG_RESTR_NIL, 24
STRONG_RESTR_PR_LAB_NIL, 24
STRONG_RESTR_PREFIX_LABEL, 24
STRONG_RESTR_PREFIX_TAU, 24
STRONG_RESTR_SUM, 24
STRONG_SUM_ASSOC_L, 24
STRONG_SUM_ASSOC_R, 24
STRONG_SUM_COMM, 24
STRONG_SUM_IDEMP, 24
STRONG_SUM_IDENT_L, 24
STRONG_SUM_IDENT_R, 24
STRONG_SUM_MID_IDEMP, 25
STRONG_UNFOLDING, 25
SYNC_BASE, 25
SYNC_def_compute, 25
SYNC_INDUCT, 25
SYNC_TRANS_THM, 26
SYNC_TRANS_THM_EQ, 26

WeakEQ Theory, 26
Definitions, 26
EPS1_def, 26
EPS_def, 26
epsilon_def, 26
RWEAK_EQUIV, 27
TRACE_def, 27
WEAK_BISIM, 27

WEAK_EQUIV_def, 27
WEAK_TRACE_def, 28
WEAK_TRANS_def, 28
Theorems, 28
EPS1_cases, 28
EPS1_ind, 28
EPS1_rules, 28
EPS1_strongind, 28
TRACE_cases, 29
TRACE_ind, 29
TRACE_rules, 29
TRACE_strongind, 29
WEAK_EQUIV, 29
WEAK_EQUIV_cases, 29
WEAK_EQUIV_coind, 30
WEAK_EQUIV_IS_WEAK_BISIM, 30
WEAK_EQUIV_rules, 30
WEAK_TRACE_cases, 30
WEAK_TRACE_ind, 30
WEAK_TRACE_rules, 31
WEAK_TRACE_strongind, 31
WEAK_TRANS_cases, 31
WEAK_TRANS_ind, 31
WEAK_TRANS_rules, 31
WEAK_TRANS_strongind, 31