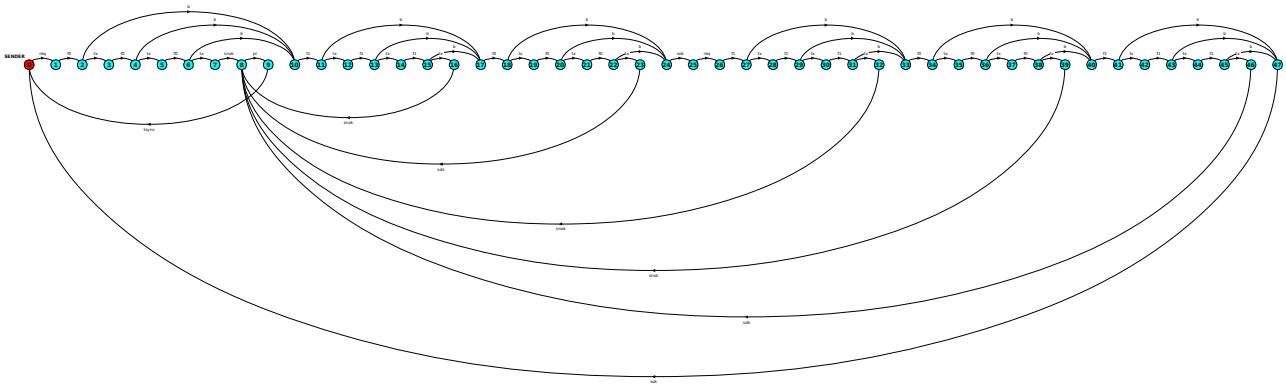
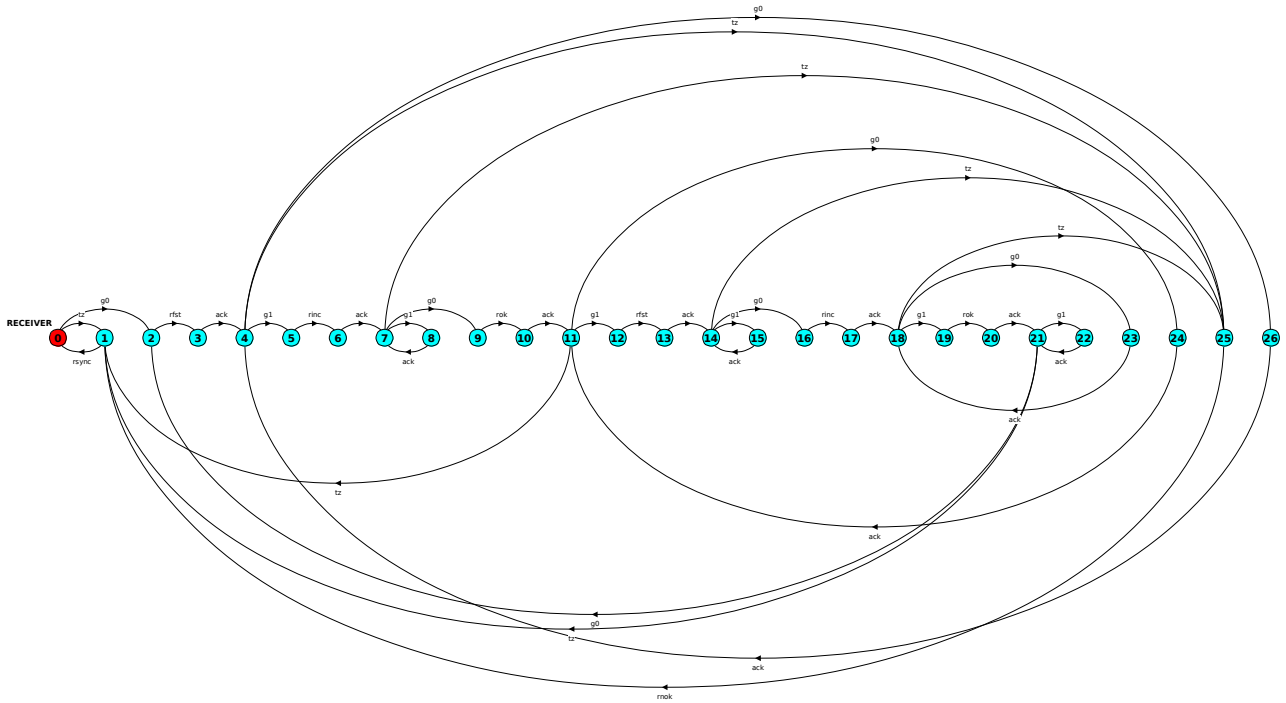
BOUNDED RETRANSMISSION PROTOCOL

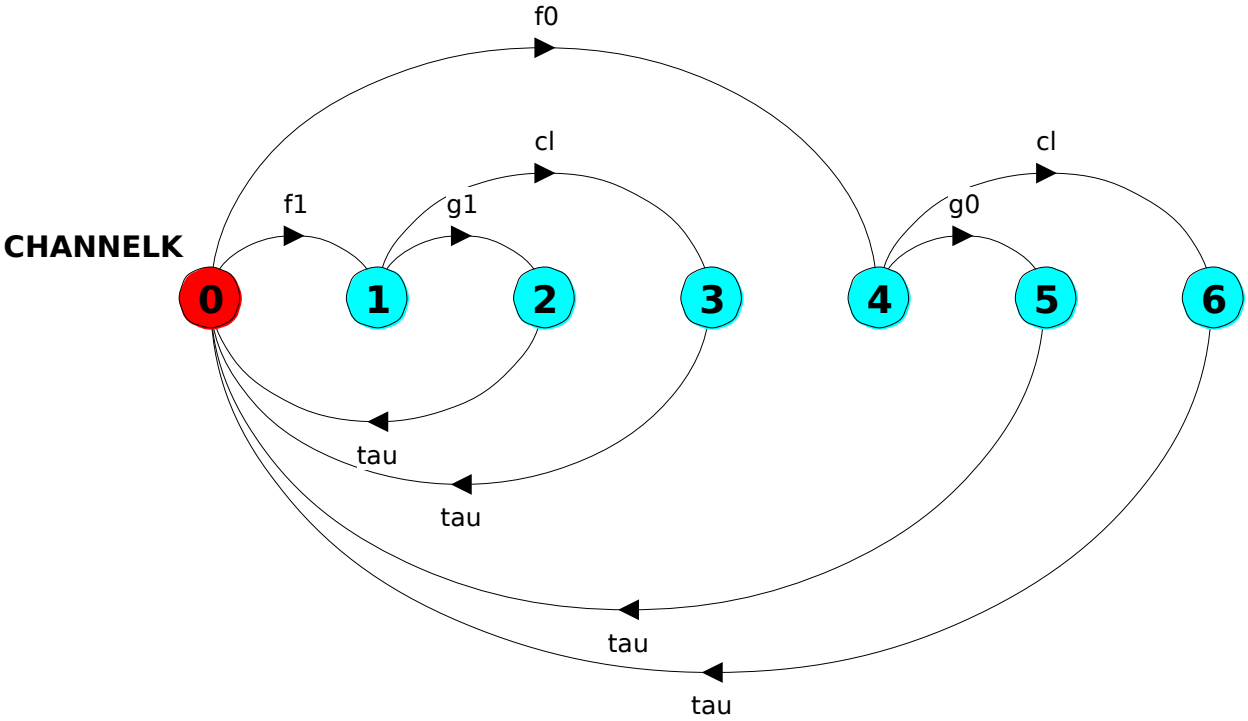
(Robert Meolic, 1999, 2013, in Slovene)

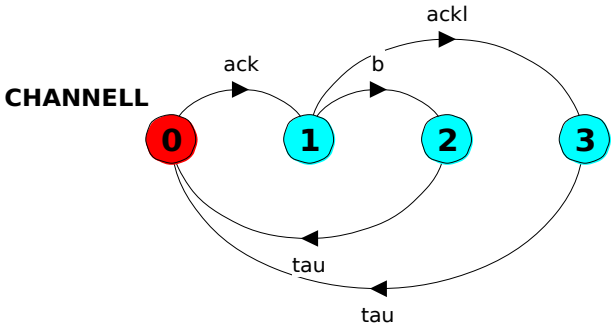
File **brp.dat** is an implementation of bounded retransmission protocol. This is correct implementation, but some incorrect variants are also given in comments. Process BRP_weak is minimal process weak equivalent to the obtained composition. Process BRP_test is minimal process testing equivalent to the obtained composition. Process BRP_trace is minimal process trace equivalent to the obtained composition.

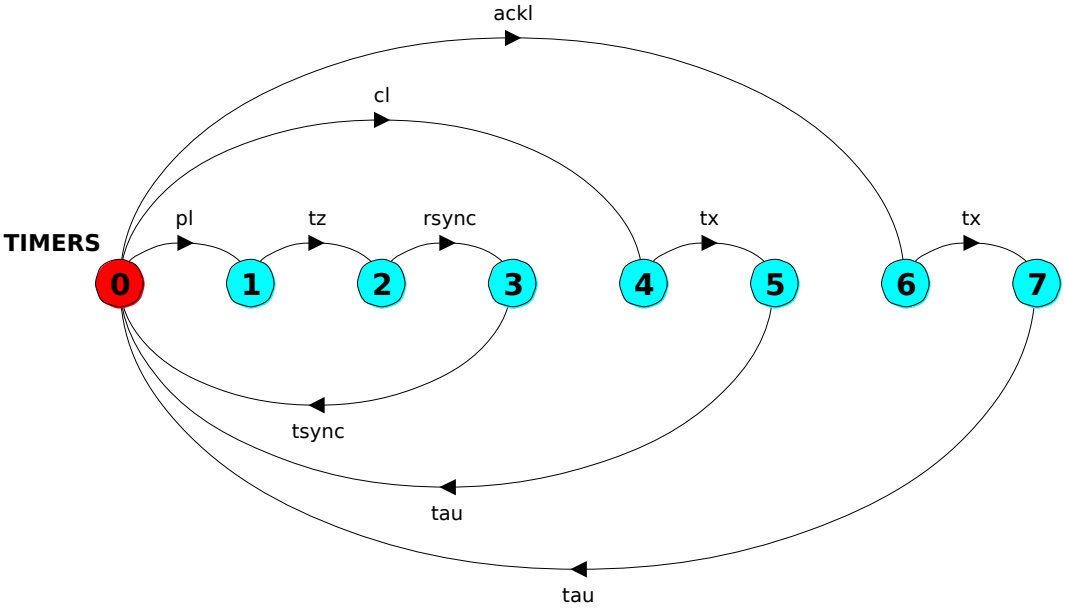
Here are LTSs used for specification and the composed LTS minimised with LTSA V3.0.

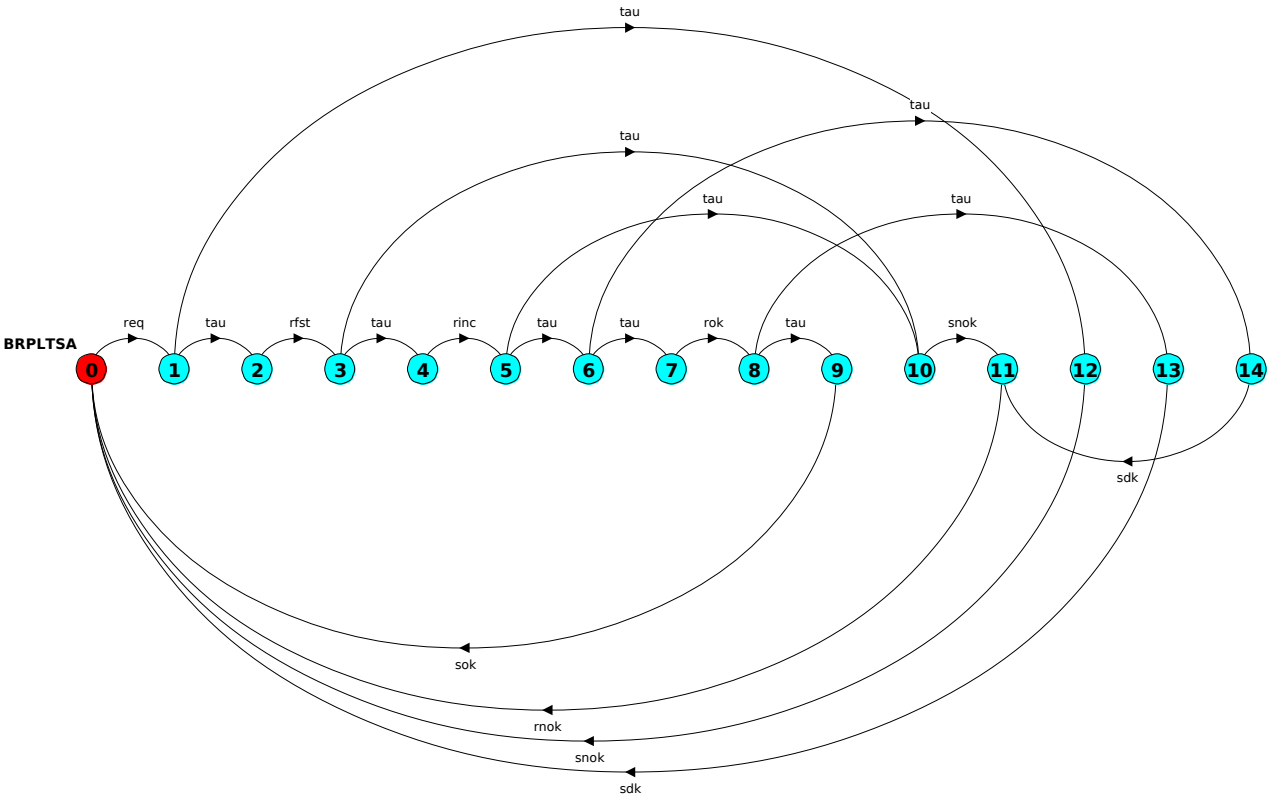





CHANNELK
f0
cl
cl
f1
g1
g0
0
1
2
3
4
5
6
tau
tau
tau
tau


CHANNELL
ackl
ack
b
0
1
2
3
tau
tau


TIMERS
ackl
cl
pl
tz
rsync
tx
tx
0
1
2
3
4
5
6
7
tsync
tau
tau

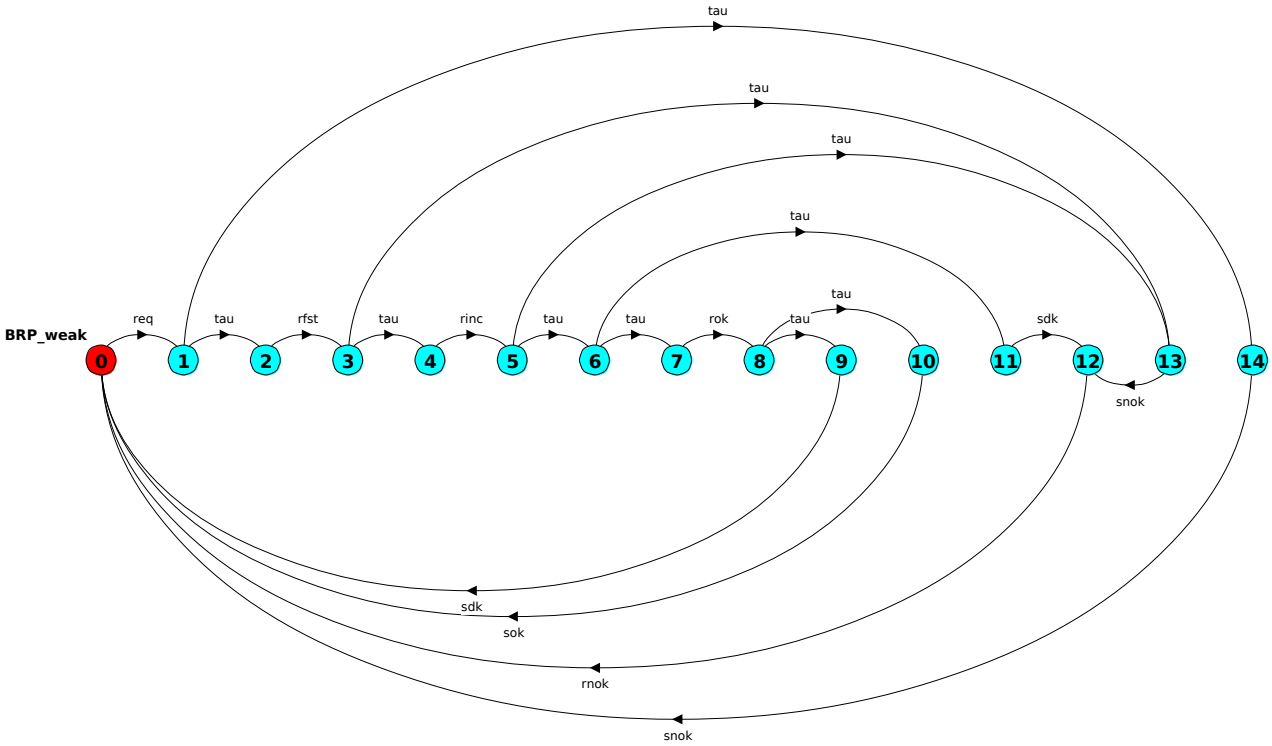


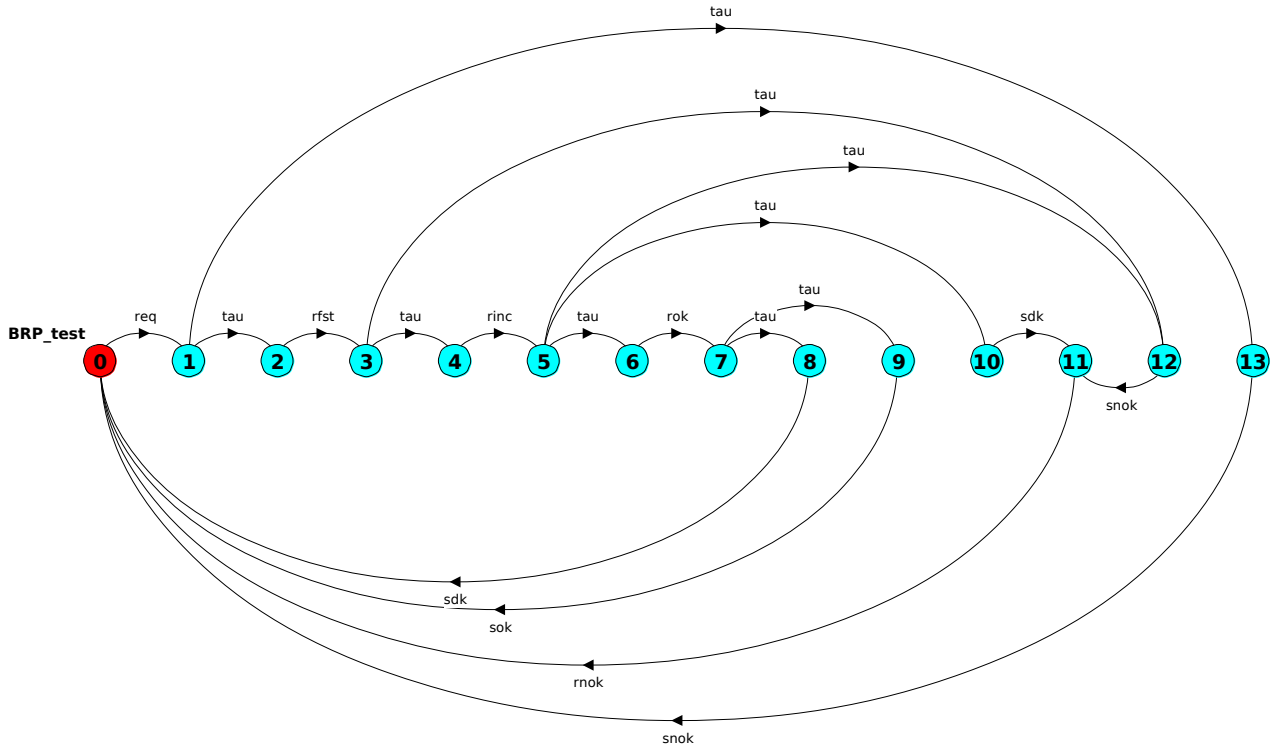
File **brp.css** enables verification with ACTLW model checking. Here is the log from EST:

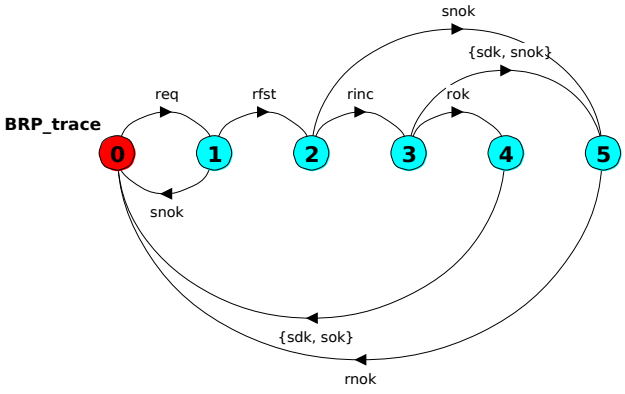
```
Efficient Symbolic Tools, 2nd Edition, Copyright (C) 2003-2013 UM-FERI
This is free software, and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute EST; type "license" for details.

Running on i686 (Linux, 3.2.0-38-generic-pae) with tcl 8.5.11 and tk 8.5.11.

Initialization of GUI package... OK
Initialization of BDD package... OK
Initialization of Process_Algebra package... OK
Initialization of Versis package... OK
Initialization of Model checking package... OK
Initialization of Strucval package... OK
Initialization of CCS package... OK
Ready.

>cd "/home/meolic/est/est-2ed/data/brp"; source "brp.tcl"; cd "/home/meolic/est/est-2ed/data"
Reading file: brp.dat
  Sort sortBRP ... OK
  Process SENDER ... OK
  Process RECEIVER ... OK
  Process CHANNELK ... OK
  Process CHANNELL ... OK
  Process TIMERS ... OK
  Process BRP_weak ... OK
  Process BRP_test ... OK
  Process BRP_trace ... OK

Reading file: brp.ccs
  Parallel composition BRP ...OK
  Property F1
    F1 = AAG AAF {REQ?};
  Property F2
    F2 = AAG [REQ?] AA[{NOT (RFST! OR RINC! OR ROK! OR RNOK!)} W {RFST!}];
  Property F3
    F3 = AAG [RFST!] AA[{NOT RFST!} U {REQ?}];
  Property F4
    F4 = AAG [ROK! OR RNOK!] AA[{NOT (RFST! OR RINC! OR ROK! OR RNOK!)} U {REQ?}];
```

```
  Property F5
    F5 = AAG [RFST! OR RINC!] AA[{NOT REQ?} U {RFST! OR RINC! OR ROK! OR RNOK!}];
  Property F6
    F6 = AAG [REQ?] AA[{NOT REQ?} U {SOK! OR SNOK! OR SDK!}];
  Property F7
    F7 = AAG [SOK! OR SNOK! OR SDK!] AA[{NOT (SOK! OR SNOK! OR SDK!)} U {REQ?}];
  Property F8
    F8 = AAG [ROK!] AA[{NOT SNOK!} U {REQ?}];
  Property F9
    F9 = AAG [RNOK!] AA[{NOT SOK!} U {REQ?}];
  Property F10
    F10 = AAG [SOK!] AA[{NOT RNOK!} U {REQ?}];
  Property F11
    F11 = AAG [SNOK!] AA[{NOT ROK!} U {REQ?}];
  Property F12
    F12 = AAG [REQ?] AA[{NOT SDK!} W {RFST!}];

COMPOSITION BRP: 1726 st, 3907 tr, 396 vis

# 1. Podatkovni paketi se prenasajo neskoncno mnogokrat
ACTL/ACTLW model checking on composition BRP
AAG AAF {REQ?} ==> TRUE

#2. Prvo sprejeto sporocilo za podatkovni paket je IFST
ACTL/ACTLW model checking on composition BRP
AAG [REQ?] AA[{NOT (RFST! OR RINC! OR ROK! OR RNOK!)} W {RFST!}] ==> TRUE

3. Med prenosom podatkovnega paketa dobi prejemnik najvec en IFST
ACTL/ACTLW model checking on composition BRP
AAG [RFST!] AA[{NOT RFST!} U {REQ?}] ==> TRUE

4. Zadnje sprejeto sporocilo za podatkovni paket je IOK ali INOK
ACTL/ACTLW model checking on composition BRP
AAG [ROK! OR RNOK!] AA[{NOT (RFST! OR RINC! OR ROK! OR RNOK!)} U {REQ?}] ==> TRUE

5. IFST in IINC nista zadnji sprejeti sporocili za podatkovni paket
ACTL/ACTLW model checking on composition BRP
AAG [RFST! OR RINC!] AA[{NOT REQ?} U {RFST! OR RINC! OR ROK! OR RNOK!}] ==> TRUE

6. Enota P ne zacne naslednjega prenosa, dokler ne dobi obvestilo
ACTL/ACTLW model checking on composition BRP
AAG [REQ?] AA[{NOT REQ?} U {SOK! OR SNOK! OR SDK!}] ==> TRUE

7. Enota P dobi le eno obvestilo za vsak podatkovni paket
ACTL/ACTLW model checking on composition BRP
AAG [SOK! OR SNOK! OR SDK!] AA[{NOT (SOK! OR SNOK! OR SDK!)} U {REQ?}] ==> TRUE

8. Enota P ne dobi INOK, ce dobi enota C IOK in obratno
ACTL/ACTLW model checking on composition BRP
AAG [ROK!] AA[{NOT SNOK!} U {REQ?}] ==> TRUE

ACTL/ACTLW model checking on composition BRP
AAG [RNOK!] AA[{NOT SOK!} U {REQ?}] ==> TRUE

ACTL/ACTLW model checking on composition BRP
AAG [SOK!] AA[{NOT RNOK!} U {REQ?}] ==> TRUE

ACTL/ACTLW model checking on composition BRP
AAG [SNOK!] AA[{NOT ROK!} U {REQ?}] ==> TRUE

9. Enota P ne dobi IDK, ce ni prenesen noben kos podatkovnega paketa
ACTL/ACTLW model checking on composition BRP
AAG [REQ?] AA[{NOT SDK!} W {RFST!}] ==> TRUE
```

File **minimization.tcl** is used for studying minimisation procedures and equivalence relations. Here are processes BRP_weak, BRP_test, and BRP_trace visualised with LTSA V3.0.


BRP_weak
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
req
tau
rfst
tau
rinc
tau
tau
rok
tau
tau
sdk
tau
tau
tau
tau
tau
snok
sdk
sok
rnok
snok


BRP_test
0
1
2
3
4
5
6
7
8
9
10
11
12
13
req
tau
rfst
tau
rinc
tau
rok
tau
tau
sdk
tau
tau
tau
tau
tau
snok
sdk
sok
rnok
snok


BRP_trace
0
1
2
3
4
5
req
rfst
rinc
rok
snok
{sdk, snok}
snok
{sdk, sok}
rnok

Here is the log from EST:

```
Efficient Symbolic Tools, 2nd Edition, Copyright (C) 2003-2012 UM-FERI
This is free software, and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute EST; type "license" for details.

Running on i686 (Linux, 3.2.0-38-generic-pae) with tcl 8.5.11 and tk 8.5.11.

Initialization of GUI package... OK
Initialization of BDD package... OK
Initialization of Process_Algebra package... OK
Initialization of Versis package... OK
Initialization of Model checking package... OK
Initialization of Strucval package... OK
Initialization of CCS package... OK
Ready.

>cd  "/home/meolic/est/est-2ed/data/brp";  source  "minimization.tcl";  cd  "/home/meolic/est/est-
2ed/data"
Reading file: brp.dat
  Sort sortBRP ... OK
  Process SENDER ... OK
  Process RECEIVER ... OK
  Process CHANNELK ... OK
  Process CHANNELL ... OK
  Process TIMERS ... OK
  Process BRP_weak ... OK
  Process BRP_test ... OK
  Process BRP_trace ... OK

Parallel composition (1): BRP...
Strong minimization of BRP... OK
Weak minimization of BRP... OK
Trace minimization of BRP... OK

COMPOSITION BRP: 1726 st, 3907 tr, 396 vis
STRONG_BRP: 508 st, 1124 tr, 124 vis
WEAK_BRP: 15 st, 20 tr, 10 vis
TRACE_BRP: 6 st, 11 tr, 11 vis
BRP_weak: 15 st, 20 tr, 10 vis
BRP_test: 14 st, 19 tr, 10 vis
BRP_trace: 6 st, 11 tr, 11 vis

Strong observational equivalence checking between STRONG_BRP and BRP... OK
Weak observational equivalence checking between WEAK_BRP and BRP... OK
Trace equivalence checking between TRACE_BRP and BRP... OK
Strong observational equivalence checking between BRP_weak and BRP... NOT EQUIVALENT
Strong observational equivalence checking between BRP_test and BRP... NOT EQUIVALENT
Strong observational equivalence checking between BRP_trace and BRP... NOT EQUIVALENT
Weak observational equivalence checking between BRP_weak and BRP... OK
Weak observational equivalence checking between BRP_test and BRP... NOT EQUIVALENT
Weak observational equivalence checking between BRP_trace and BRP... NOT EQUIVALENT
Testing equivalence checking between BRP_weak and BRP... OK
Testing equivalence checking between BRP_test and BRP... OK
Testing equivalence checking between BRP_trace and BRP... NOT EQUIVALENT
Trace equivalence checking between BRP_weak and BRP... OK
Trace equivalence checking between BRP_test and BRP... OK
Trace equivalence checking between BRP_trace and BRP... OK
Weak observational equivalence checking between WEAK_BRP and BRP_weak... OK
Trace equivalence checking between TRACE_BRP and BRP_trace... OK
Testing equivalence checking between STRONG_BRP and BRP_test... OK
Testing equivalence checking between WEAK_BRP and BRP_test... OK
Testing equivalence checking between TRACE_BRP and BRP_test... NOT EQUIVALENT
```