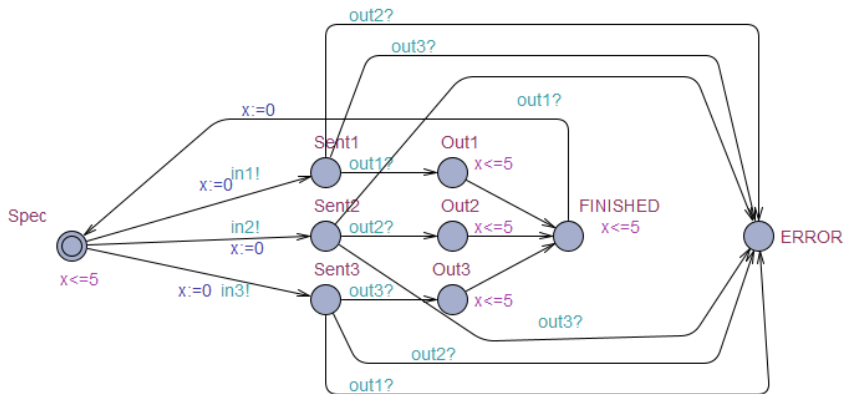
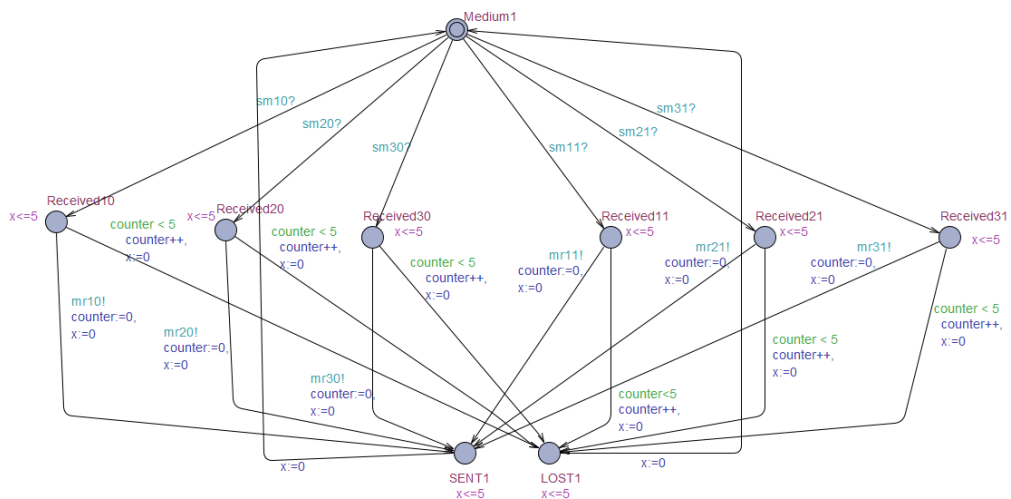


1. Najprv vyriešime problém, že sa automat zo stavov nemusí vôbec pohnúť preč. Urobíme to tak, že všetky stavy, z ktorých vychádza akcia tau alebo akcia poslania do kanála, dáme časové obmedzenie a následne čas vynulujeme. Takto ošetríme všetky potrebné stavy, tým, že sa do 5 časových jednotiek z miesta musia pohnúť.

Príklad:

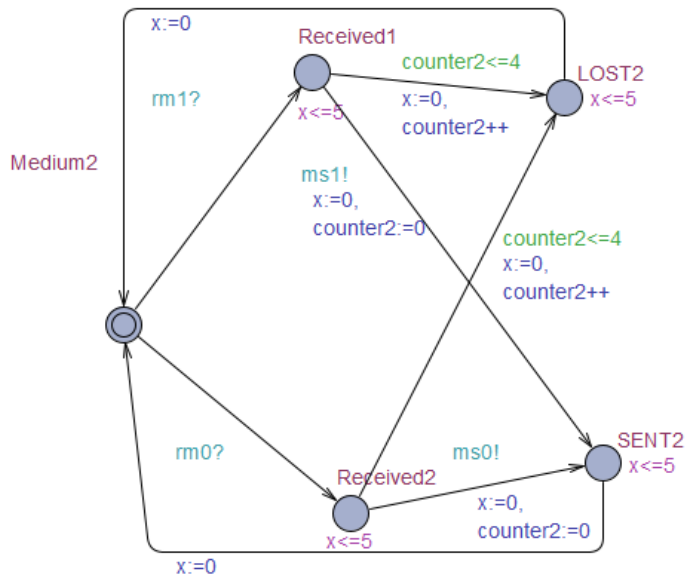


Následne chceme vyriešiť, aby sa správa v médiu 1 nestrácala donekonečna, ale nakoniec bola nútená sa poslať receiverovi. Urobíme to pomocou premennej counter, ktorú najprv nastavíme na 0, následne jej hodnotou podmienime transition do stavu, kde je správa stratená. Povolíme tak len 5x po sebe tú istú správu stratiť, potom už je médium nútené ju preposlať.

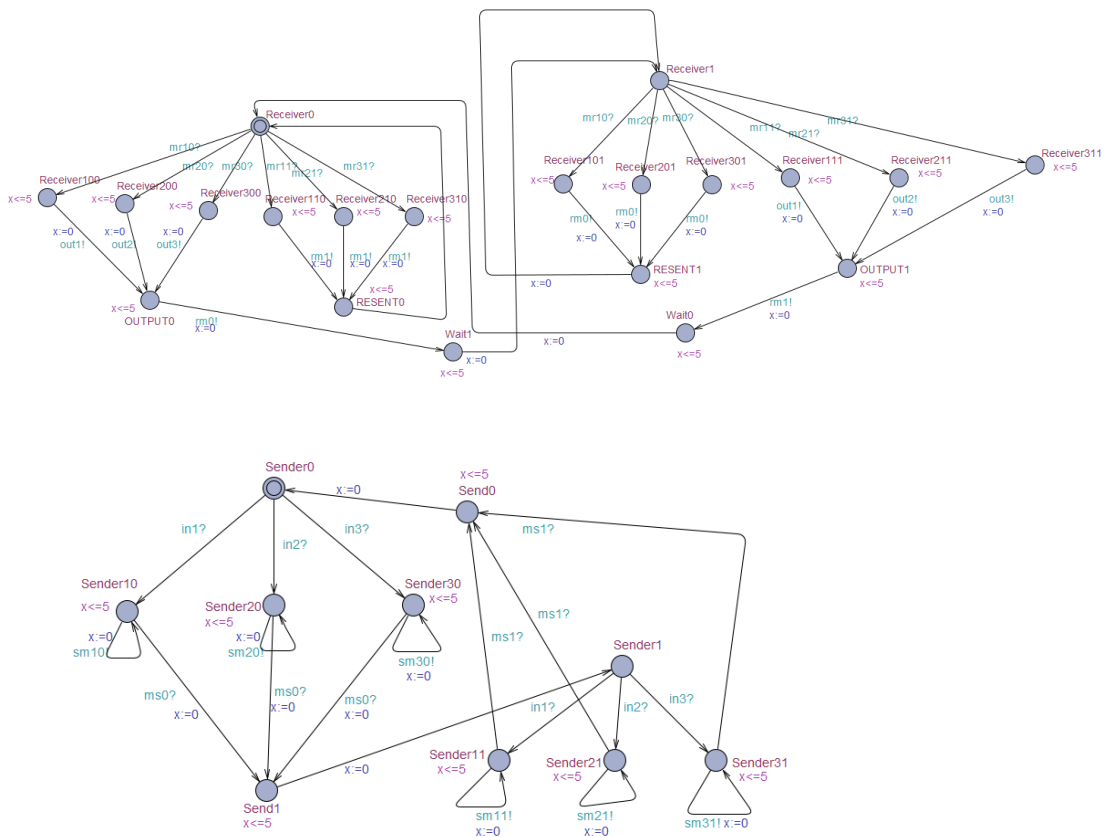


Pri prechode do stavu LOST1 sa kontroluje veľkosť counteru a inkrementuje sa jeho veľkosť (counter je int variable) a v prechode, keď správu už pošleme sa counter resetuje na 0.

Aby sme si boli istí, že sa potvrdzovacia správa dostane naspäť k senderovi, využijeme ďalšiu premennú counter2, ktorá rovnakým mechanizmom kontroluje strácanie správy médiom2



Ešte Sender a Receiver s časovými obmedzeniami na stavy:



Deklarácia premenných:

```
int counter = 0;
int counter2 = 0;
chan rm0, rml, ms0, ms1, out1, out2, out3, sm10, mrl0, mrl1, mr20, mr21, mr30, mr31, sm11, sm20, sm21, sm30, sm31, in1, in2, in3;
lock x;

S = Sender(sm10, sm20, sm30, sm11, sm21, sm31, ms0, ms1, in1, in2, in3, x);
M1 = Medium1(mrl0, mrl1, mr20, mr21, mr30, mr31, sm10, sm11, sm20, sm21, sm30, sm31, x, counter);
R = Receiver(mrl0, mrl1, mr20, mr21, mr30, mr31, rm0, rml, out1, out2, out3, x);
M2 = Medium2(rm0, rml, ms0, ms1, x, counter2);
T = Tester(in1, in2, in3, out1, out2, out3, x);

system S, M1, R, M2, T;
```

Overíme pôvodný a upravený automat formulou $T.Sent1 \rightarrow T.Out1$ (teda, či sa zo stavu, keď pošleme správu 1 eventuálne dostane na výstup). To isté pre správu 2 aj správu 3.

```
T.Sent2 --> T.Out2
T.Sent1 --> T.Out1
T.Sent3 --> T.Out3
```

Zatiaľ čo pred úpravou sme formuly nespĺnili, po úprave už áno

```
T.Sent1 --> T.Out1
Verification/kernel/elapsed time used: 0.094s / 0.016s / 0.116s.
Resident/virtual memory usage peaks: 18,720KB / 64,928KB.
Property is satisfied.
T.Sent2 --> T.Out2
Verification/kernel/elapsed time used: 0.109s / 0.015s / 0.131s.
Resident/virtual memory usage peaks: 18,840KB / 65,052KB.
Property is satisfied.
T.Sent3 --> T.Out3
Verification/kernel/elapsed time used: 0.094s / 0s / 0.1s.
Resident/virtual memory usage peaks: 18,844KB / 65,056KB.
Property is satisfied.
```

Len pre kontrolu si overíme, či sa náš automat môže dostať do deadlocku:

```
A[] not deadlock
Verification/kernel/elapsed time used: 0.25s / 0s / 0.238s.
Resident/virtual memory usage peaks: 19,392KB / 66,436KB.
Property is satisfied.
```