

## 10. Domáca úloha Nikolaj Kniha

1. V systéme CAAL overte, že Alternating bit protocol, ktorý ste predtým napísali, môže divergovať. Upravte ho, aby nemohol a overte toto formulou, ako i slabu bisimuláciu so špecifikáciou.

Upravný ABP, ktorý nebude divergovať. Mali sme 2 možnosti, buď úplne vymazať možnosť strácania správ alebo obmedziť množstvo stratení. Obmedzíme teda iba na 1 stratu pomocou ML1.

```

S = S0;
S0 = in1.S10 + in2.S20 + in3.S30;
S10 = 'sm10.(ms0.S1 + S10);
S20 = 'sm20.(ms0.S1 + S20);
S30 = 'sm30.(ms0.S1 + S30);

S1 = in1.S11 + in2.S21 + in3.S31;
S11 = 'sm11.(ms1.S0 + S11);
S21 = 'sm21.(ms1.S0 + S21);
S31 = 'sm31.(ms1.S0 + S31);

ML1 = sm10.('mr10.ML1 + M1) + sm20.('mr20.ML1 + M1) + sm30.('mr30.ML1 + M1) +
      sm11.('mr11.ML1 + M1) + sm21.('mr21.ML1 + M1) + sm31.('mr31.ML1 + M1);

M1 = sm10.('mr10.ML1) + sm20.('mr20.ML1) + sm30.('mr30.ML1) +
      sm11.('mr11.ML1) + sm21.('mr21.ML1) + sm31.('mr31.ML1);

M2 = rm0.('ms0.M2 + M2) + rm1.('ms1.M2 + M2);

R = R0;
R0 = mr10.'out1.'rm0.R1 + mr20.'out2.'rm0.R1 + mr30.'out3.'rm0.R1 +
      mr11.'rm1.R0 + mr21.'rm1.R0 + mr31.'rm1.R0;

R1 = mr11.'out1.'rm1.R0 + mr21.'out2.'rm1.R0 + mr31.'out3.'rm1.R0 +
      mr10.'rm0.R1 + mr20.'rm0.R1 + mr30.'rm0.R1;


Protokol = (S | ML1 | M2 | R) \
           {sm10, sm11, sm20, sm21, sm30, sm31, mr10, mr11, mr20, mr21, mr30, mr31, rm0, rm1, ms0, ms1};


```

```


Specifikacia = in1.'out1.Specifikacia + in2.'out2.Specifikacia + in3.'out3.Specifikacia;


```

	475 ms	Specifikacia ≈ Protokol
---	--------	-------------------------

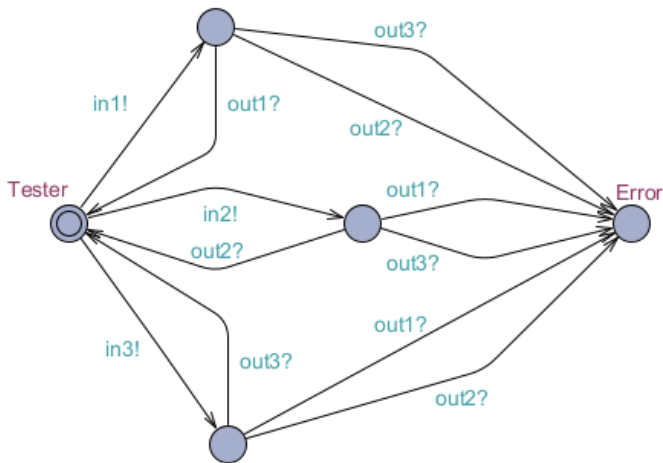
	51 ms	Protokol ≡ <in1>X X max= <tau>X
---	-------	------------------------------------

Pôvodné ABP, ktoré dokáže stále strácať správy:

	301 ms	Specifikacia ≈ Protokol
---	--------	-------------------------

	51 ms	Protokol ≡ <in1>X X max=<tau>X
---	-------	-----------------------------------

2. V UPPAALe napíšte automat tester, ktorý posiela Alternating bit protocolu správy a zároveň ich následne prijíma. Ak po správe napr. m1 prijme správu inú ako m1, dostane sa do stavu Error (podobne pre ostatné správy). Ukážte, či váš protokol spolu s testom vie dosiahnuť stav Error (pozrite si v tutoriály ako sa to zisťuje). Ak je protokol dobre napísaný, nemal by sa dať dosiahnuť. Následne pokážte Receivera tak, aby sa stav Error dal dosiahnuť.



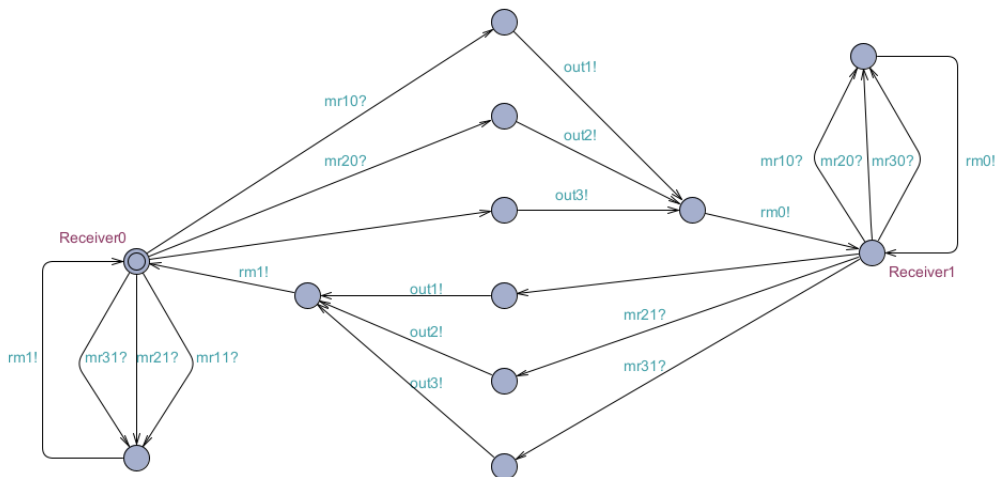
Overview

```
E<> tester.Error
```

E<> tester.Error  
Property is not satisfied.

Funguje správne, do Erroru sa nedostane.

Pokážme Receiver vymazaním mr10? a mr11?, teda nebude čakať na prijatie správy, ale rovno odošle správu out3! aj keď pôjde o inú správu.



```
E<> tester.Error
```

E<> tester.Error  
Property is satisfied.