12. Domáca úloha

Formálne metódy tvorby softvéru

Andrea Smiešna

V CAALe napíšte Alternating bit protocol a časové obmedzenia, zachovajte možnosť jeden raz stratiť správu. Overte pomocou formúl (vrátane tých, ktoré vedia špecifikovať čas - pozrite syntax):
1. ak sa správa m1 objavý na vstupe, tak sa (ako ďalšia viditeľná akcia) raz objavý na výstupe m1,
2. ak neobmedzíme počet strácaní, tak toto neplatí - overte formulou,
3. ak sa správa m1 objavý na vstupe, tak sa ako ďalšia viditeľná akcia nemôže objaviť na výstupe správa m2,

Alternating Bit Protocol zapísaný v systéme CAAL, ktorý môže raz stratiť správu.

```
Sender   = Sender0;
Sender0  = ina.3.Sendera0 + inb.3.Senderb0 + inc.3.Senderc0;
Sendera0 = 'sma0.(ms0.Sender1 + 3.Sendera0);
Senderb0 = 'smb0.(ms0.Sender1 + 3.Senderb0);
Senderc0 = 'smc0.(ms0.Sender1 + 3.Senderc0);
Sender1  = ina.3.Sendera1 + inb.3.Senderb1 + inc.3.Senderc1;
Sendera1 = 'sma1.(ms1.Sender0 + 3.Sendera1);
Senderb1 = 'smb1.(ms1.Sender0 + 3.Senderb1);
Senderc1 = 'smc1.(ms1.Sender0 + 3.Senderc1);
Medium1  = 1.sma0.('mra0.Lost) +
           1.smb0.('mrb0.Lost) +
           1.sma1.('mra1.Lost) +
           1.smb1.('mrb1.Lost) +
           1.smc0.('mrc0.Lost) +
           1.smc1.('mrc1.Lost);
Lost  = 1.sma0.('mra0.Lost + Medium1) +
           1.smb0.('mrb0.Lost + Medium1) +
           1.sma1.('mra1.Lost + Medium1) +
           1.smb1.('mrb1.Lost + Medium1) +
           1.smc0.('mrc0.Lost + Medium1) +
           1.smc1.('mrc1.Lost + Medium1);
```

```
Receiver = Receive0;

Receive0 = mra0.'outa.2.'rm0.Receive1 +
           mrb0.'outb.2.'rm0.Receive1 +
           mrc0.'outc.2.'rm0.Receive1 +
           mra1.'rm1.Receive0 +
           mrb1.'rm1.Receive0 +
           mrc1.'rm1.Receive0;

Receive1 = mra0.'rm0.Receive1 +
           mrb0.'rm0.Receive1 +
           mrc0.'rm0.Receive1 +
           mra1.'outa.'rm1.Receive0 +
           mrb1.'outb.'rm1.Receive0 +
           mrc1.'outc.'rm1.Receive0;
```

```
Medium2  = rm0.('ms0.Medium2 + Medium2) + rm1.('ms1.Medium2 + Medium2);

Protokol = (Sender | Lost | Medium2  | Receiver) \
           {sma0,sma1,smb0,smb1,mra0,mra1,mrb0,mrb1,rm0,rm1,ms0,ms1, smc0, smc1,mrc0, mrc1};
```

Doplnili sme časové obmedzenia. Keď sa správa a, b alebo c odošle, dostane sa na vstup (in), tak predtým ako ju protokol odošle, počká 3 časové jednotky. Ak Senderovi nepríde potvrdenka o prijatí správy, potom počká 3 časové jednotky a správu odošle znova. 1 časovú jednotku čaká aj Medium1 a Lost predtým ako prijme správu od Sendera.

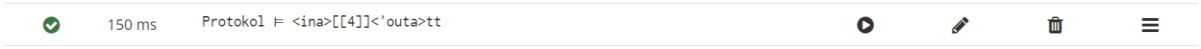
1.

Pomocou formuly overíme, že ak sa správa a objaví na vstupe, tak sa raz objaví na výstupe a.

V systéme CAAL v časti "verify" zapíšeme formulu, ktorá reprezentuje špecifikáciu protokolu. Teda ak sa určitá správa dostala na vstup, potom sa po čase dostane táto správa na výstup. Keďže protokol po prijatí správy čaká 3 časové jednotky a ak sa správa stratila, potom sa znovu odošle tá istá správa po 3 časových jednotkách.

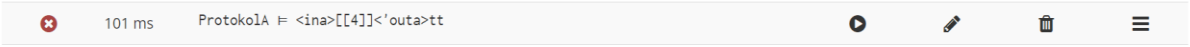
Použili sme formulu, ktorá vie špecifikovať čas (weak universal timed modality), pomohli sme si syntaxou systému CAAL. Formula reprezentuje všeobecný kvantifikátor, teda zakaždým musí platiť špecifikácia protokolu.

Touto formulou sme skontrolovali Protokol, ktorý môže správu stratiť najviac raz. Takýto upravený protokol má spĺňať formulu napríklad pre správu 1 - $< ina > [[4]] < 'outa >$.



2.

Pomocou formuly overíme, že ak neobmedzíme počet strácaní, potom neplatí predchádzajúca časť príkladu. Použijeme rovnakú formulu ako v časti 1. Skontrolujeme protocol ProtokolA, ktorý mohol divergovať, čiže stale strácať správy, ten by nemal spĺňať formulu.



3.

Pomocou formuly overíme, že ak sa správa a objaví na vstupe, tak sa nemôže na výstupe objaviť správa b.

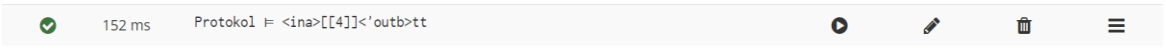
4. pokazte Receiver tak, aby sa m2 mohlo objaviť na výstupe po m1 na vstupe a overte,
5. napíšte formulu, ktorá zaručí, v akom najhoršom čase sa správa objaví na výstupe po jej načítaní,
6. upravte predchádzajúcu formulu (jej časové obmedznie) tak aby nebola splnená,
7. napíšte formulu, ktorá vyjadrí po akom čase sa môže načítať ďalšia správa.

4.

Pokazíme Receiver tak, aby sa správa b mohla objaviť na výstupe po a na vstupe. Receiver0 sme pokazili tak, že sme prepísali outa namiesto outb. Podobne by sme to spravili aj pre Receiver1 a pre zvyšné správy.

```
Receive0 = mra0.'outb.'rm0.Receive1 +
           mrb0.'outa.'rm0.Receive1 +
           mrc0.'outc.'rm0.Receive1 +
           mra1.'rm1.Receive0 +
           mrb1.'rm1.Receive0 +
           mrc1.'rm1.Receive0;
```

Overíme pomocou formuly z predchádzajúcej časti.

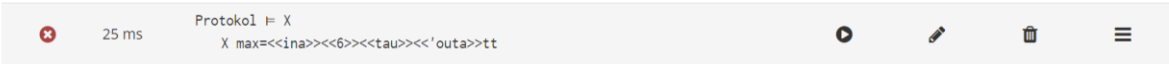


5.

Konkrétne správa sa v protokole môže objaviť na výstupe po jej načítaní, keď prejde 5 časových jednotiek. Toto je najhorší čas.

6.

Podobne ako v predchádzajúcej časti využijeme tú istú formulu. Časovú jednotku však zvýšime o 1. Takúto formulu už protokol nespĺňa.



7.

Pomocou formuly overíme, poslanie trvá najhoršie 5 časových jednotiek.

Trvá 2 časové jednotky, kým sa pošle ďalšia správa.