8. Zápis a verifikace paralelních systémů

1. P/S

P(P1, P2) paralelní provádění P1 a P2
S(P1, P2) sériové provádění P1, P2

(a*b+c*d)+(a/e)

S(P(S(P(t1=a*b, t2=c*d), t4=t1+t2), t3=a/e), r=t4+t3)

2. cobegin/coend

Dijkstra: parbegin/cobegin

cobegin
P1|P2|P3;
coend

co
P1 // P2 // P3;
oc

Lamport: **< A; B; C; >** atomická operace

3. fork/join

fork *label* spuštění procesu od návěští
quit ukončení procesu
join *m,label* m=m-1, když m==0 skok na návěští

```
m = 3;
fork P1; fork P2;
join m,P3; quit;
P1: ...; join m,P3; quit;
P2: ...; join m,P3; quit;
P3: ...
```

Verifikace

1. Formální logika - Owicki, Gries (1975), Lamport (1977), Pnuelli - LTL (1977)
2. Modelování

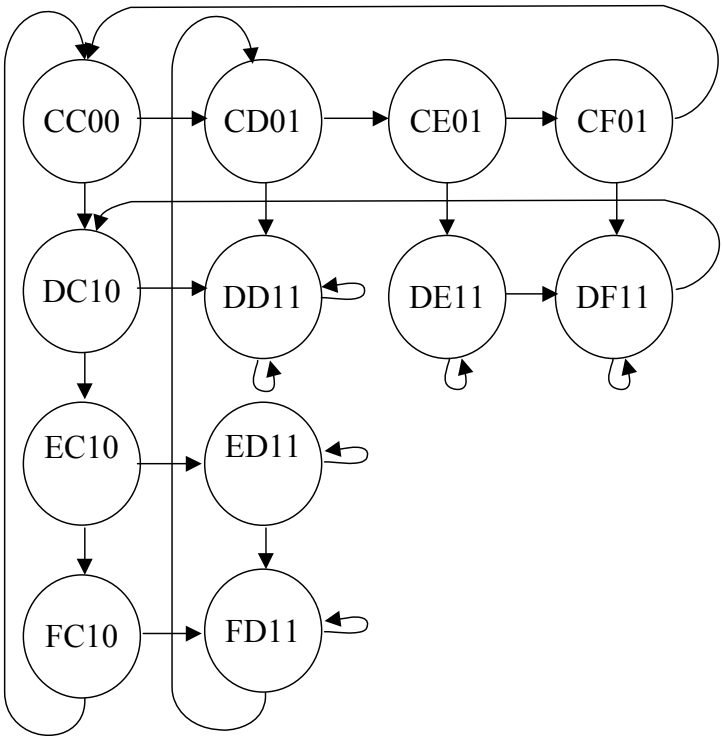
Princip:
1. Specifikace (tvrzení, globální invarianty)
2. Model
3. Verifikace, zda model odpovídá specifikaci

Nástroje:
- CSP
- Petriho sítě
- spin/PROMELA
- verifikační systémy (LTL)

Stavový prostor

Kripkeho model (S, R, L) - stavový prostor paralelního systému
S - konečná množina stavů
R - množina přechodů mezi stavy S
L - značení, definuje hodnotu atomických tvrzení pro každý stav
cesta - posloupnost $\sigma = s_1, s_2, s_3, \ldots \quad (s_i, s_{i+1}) \in R$
Problém: počet stavů je exponenciální - redukce

```
P                              Q
    shared int flag1 = 0, flag2 = 0;
    while (1) {                while (1) {
C:      flag1 = 1;                 flag2 = 1;
D:      while (flag2) ;            while (flag1) ;
E:      kritická sekce             kritická sekce
F:      flag1 = 0;                 flag2 = 0;
        výpočet                    výpočet
    }                          }
```



Značení - (příkaz, příkaz, stav flag1, stav flag2)
Konstrukce - Začneme od počátečního stavu a přidáváme všechny stavy, do kterých se může systém dostat, přičemž stavy se stejným značením tvoří jeden uzel. Z každého uzlu vede cesta dvěma směry (máme 2 procesy, pro více vícerozměrný prostor), směr vpravo = postup Q, dolů = postup P.

Bezpečnost = neexistuje cesta do stavu označeného EE**
Živost = neexistuje nekonečný cyklus obsahující alespoň jeden vertikální a současně alespoň jeden horizontální přechod a neprocházející kritickou sekcí v původním grafu, ani v grafu vzniklém po zablokování jednoho procesu v sekci výpočet (reprezentujeme změnou přechodů C->D*** na C->C***).

Uváznutí a stárnutí = nekonečný cyklus neprocházející kritickou sekci při postupu v obou procesech (obou směrech)
Blokování = nekonečný cyklus, kdy zůstává jeden proces v synchronizaci a druhý stojí v sekci výpočet (před příkazem C).