

## Nyitott specifikáció és aszinkronitás transzformáció.

Néha jól jön, ha szétszétjuk a feladatot, mert a párhuzamos feldolgozás hatékonyabb.

$$S_1 = \{se : x := x \oplus d, \text{ ha } bu \parallel \dots, re : vs, x := f(vs, x), g(x), \text{ ha } b(x) \wedge bv \parallel \dots\}$$

ahol  $bu$  egy  $x$ -től nem függő logikai feltétellel  $x$  pedig közös változó, ezért a folyamatok óhatatlanul várattják egymást (ha legfeljebb egy közös változó legfeljebb egyszer szerepel az értékadásokban).

Át szeretnénk alakítani valami jobbra, ehhez az alábbi három feltételnek kell teljesülnie:

- $b(x) \Rightarrow b(x \oplus d)$  – monotonitás  
 $x$  egy adott állapotban van, nem függhet a feldolgozási képessége és hajlandósága attól, hogy az olvasó lassabb volt mint a küldő és közben nőtt  $x$  információtartalma.
- $b(x) \Rightarrow (f(vs, x) = f(vs, x \oplus d))$  – ha én nem használtam fel  $x$ -et, akkor ha jött közben új adat, akkor se változzon.
- $b(x) \Rightarrow (g(x \oplus d) = g(x) \oplus d)$  – kommutativitás. Az a két lépés, hogy törölök és hogy jön az új, felcserélhető.

Megjegyzés: ez most nincs benne a jegyzetben, lehet félreértem, de így gondolom én ezt:  $se$  a küldő folyamat, ez ha éppen olyanja van ( $bu$ ), ráír  $x$  csatornára egy  $d$  adatot.  $re$  a fogadó folyamat, ez, ha a csatorna olvasható ( $b(x)$ ) és ha épp ráér fogadni ( $bv$ ), akkor fogad. Ez azt jelenti, hogy a saját kis feldolgozott változóját ( $vs$ ) módosítja az aktuális legfőbb értékekkel ( $f(vs, x)$ ), majd  $x$ -ből törli ezt az elemet ( $g(x)$ ).

Ha ezek teljesülhetnek, a fenti program átalakítható így:

$$S_2 = \left\{ \begin{array}{l} se : x, c := x \oplus d, (c; d) \text{ ha } bu \parallel \dots, \\ re : vs, x, y := fs(vs, y), g(x), g(y) \text{ ha } b(y) \wedge bv \parallel \dots, \\ ch : c, y := lorem(c), y \oplus c.lov \text{ ha } |c| > 0 \end{array} \right\}$$

Itt  $c$  egy új csatornaváltozó, valamint  $y$  is, meg bejött még az új  $b(y)$  feltétel, ami nyilván  $y$ -től függ. Csak  $x$  az ami  $se$  és  $re$  között osztott változó, viszont mivel az  $y$ -re késleltetve jövő adatot használom mindenhol ( $x$  csak magára hivatkozik, senki se használja őt, hanem  $y$ -t), ezért  $x$ -et akár ki is törölhetném.  $x$  inentől csak a könnyebb bizonyíthatóság miatt van, mint segédváltozó, de a lényegi kommunikációt aszinkron, késleltetett módon  $c$  és  $y$  intézi.

A program  $\mapsto$  tulajdonságai így nem maradnak meg, de a többi ( $\triangleright_s, \hookrightarrow_s, inv_s$ ) igen. (amúgy is csak  $\hookrightarrow$  számításához szoktuk használni, amúgy nem fontos kikötés az  $\mapsto$ .)

Egész pontosan az alábbi módon maradnak meg a tulajdonságok. Ha az eredeti programban  $P$  és  $Q$  között  $\triangleright_s$  vagy  $\hookrightarrow_s$  kapcsolat volt, akkor az új programban  $P'$  és  $Q'$  lesz. Hasonlóan, ha  $P$  invariáns volt, akkor  $P'$  lesz az.  $P'$  pedig így számítható  $P$ -ből ( $Q'$  hasonlóan):

$$P' = P^{x \leftarrow y \oplus c}$$

hiszen lényegében erre cseréltem ki  $x$ -et. Amúgy még van egy invariáns az új, átalakított programban, ami pont erről szól:

$$(x = y \oplus c) \in inv_{s_2}$$

Ezt a változtatást sok processzoron érdemes meghúzni, így a rendszerben kevesebben fognak várakozni. Ára ennek, egy új csatornaváltó bevezetése vót.