Tartalomjegyzék

Tétel 1: Feladat definíciója (példával)

Manapság sokszor nem elég gyors, nem elég hatékony csupán szekvenciális megoldásokat alkalmaznunk: szükség van párhuzamos, elosztott programozásra. A korábban tanult elő- és utófeltételek az ilyen rendszerek (például operációs rendszer) specifikálására azonban már nem elégségesek, ezért általánosítjuk a feladat fogalmát.

Programozási feladat

A egy állapottér, B egy paramétertér. Mindkettő típusértékhalmaz(ok) direktszorzata.

Definíció: $F \subseteq B \times (\times_{i\in[1..3]}\mathcal{P}(\mathcal{P}(A) \times \mathcal{P}(A)) \times_{i\in[1..4]} \mathcal{P}(\mathcal{P}(A)))$

A feladat paraméterekhez rendel specifikációs feltételeket. A feladatot megoldják azok a programok, amelyek minden $b \in B$ paraméter esetén megfelelnek ezeknek a feltételeknek.

A paramétertér viselkedése megegyezik a klasszikus modellel: célja, hogy a potenciálisan végtelen sok kezdőállapotból induló programok helyett csak egy bizonyos (a paraméterek által leszűkített) halmazon vizsgálódjunk.

Például az $a + b = c$ értéket kiszámoló feladat esetén a és b értéke paraméterek által lenne meghatározva, de c értéke nem.

A feladat reláció nem egy függvény: egy $b \in B$ értékhez több reláció-hetes is tartozhat.

Specifikációs feltételek

Legyen $h \in F(b)$. Ha $F(b)$ egyelemű, akkor h helyett b-nek jelöljük.

Átmenetfeltételek

- $P \triangleright_h Q$, "P, feltéve, hogy nem Q", unless
  - Ha $P \wedge \neg Q$ teljesül, akkor Q érintése nélkül tilos közvetlenül a $\neg P \wedge \neg Q$ állapotba kerülni
- $P \mapsto_h Q$, "P biztosítja Q", ensures
  - $P \triangleright_h Q$ azzal kiegészítve, hogy egy lépésben kell, hogy legyen lehetősége a programnak $P \wedge \neg Q$-ból Q-ba jutnia
  - Tehát előbb-utóbb ennek az átmenetnek be kell következnie, méghozzá egyetlen lépésben lehetségesnek kell lennie
- $P \hookrightarrow_h Q$, "P-ből elkerülhetetlen Q", leads-to
  - Előbb-utóbb (1, vagy több lépésben) $P \wedge \neg Q$ állapotból a program Q állapotba fog kerülni
- $P \in inv_h$, "P invariáns"
  - $P \in inv_h \equiv inv_h P$
  - P a kezdeti értékadást követően teljesül és P igazsághalmazából minden állapotátmenet a P igazsághalmazába visz át (tehát a programnak nem szabad elhagyja P-t)

- $P \in TERM_h$, "Q-ból a program biztosan fixpontba jut"
  - $P \in TERM_h \equiv P \hookrightarrow FP_h$

Peremfeltételek

- $R \in FP_h$, "R teljesül fixpontban"
  - $R \in FP_h \equiv FP_h \Rightarrow R$
  - Szükséges feltételek arra vonatkozóan, hogy mi teljesüljön fixpontban
- $Q \in INIT_h$, "Q igaz kezdetben"
  - Elégséges, ha egy program csak az itt definiált állapotokból indulva működik helyesen

Példa

Elemenkénti feldolgozás feladata.

$A = \overset{X}{x} \times \overset{Y}{y}$ és $B = \overset{X}{x'}$

- $Q = (x = x')$
- $Q \in INIT_h$
- $Q \hookrightarrow FP_h$ (vagy $Igaz \hookrightarrow FP_h$)
- $FP_h \Rightarrow (y = f(x'))$

Feladat finomítása (szigorítása, általánosítása)

F_1 finomítása F_2-nek, ha minden F_1-et megoldó S program megoldja F_2-t is.

Másképp: $(\forall S : S$ megold $F_1 \Rightarrow S$ megold $F_2) \Longleftrightarrow F_1$ finomítása F_2-nek

Feladatok ekvivalenciája

Két feladat ekvivalens, ha egymásnak finomításai.