

# A funkcionális programozásról szóló előadások vázlata

## 1. előadásblokk

Utolsó módosítás: 2005. nov. 5. Hanák Péter

A **2005/06-os tanév őszi félévében** a Deklaratív programozás c. tárgy funkcionális programozásról tartott előadásai – G. Smolka német nyelvű jegyzetét<sup>1</sup> követve – a következő témákról szóltak.

**Az előadásokon csak nagyon röviden** foglalkoztunk, foglalkozunk az SML nyelv szintaxisával és szemantikájával: szavakkal és mondatokkal, valamint ezek egy- és két-dimenziós ábrázolásával; értékekkel és környezetekkel; továbbá kifejezések, eljárás hívások, deklarációk és programok végrehajtásával; a szemantikai egyenlőség kérdésével.

G. Smolka előadásjegyzete alapján összefoglaló készült a témakörrel *Az SML egy résznyelvének szintaxisa és szemantikája* címen, amely letölthető a tárgy honlapjáról: <http://dp.iit.bme.hu/dp05a/index.html#SMLr1szintax>. Ezt a témakört a hallgatóknak önállóan kell feldolgozniuk.

---

<sup>1</sup>Lásd <http://www.ps.uni-sb.de/~smolka/programmierung.html>.

## 1. előadás, okt. 3.

1. Funkcionális vagy applikatív programozás: az elnevezés magyarázata.
2. SML-értelmezők és fordítóprogramok: mosml, sml-nj, Alice, Poly/ML. A *read-eval-print* ciklus.
3. Egyszerű SML-programok és elemeik: deklaráció, kifejezés; azonosító, állandó, operátor, kulcsszó; kötés.
4. Típus, típusmegkötés, típuslevezetés. `int`, `real`, `string`.
5. SML-értelmezők használata. A pontosvessző (`:`) szerepe. Egy azonosító újradeklarálása. Az `it` eredményazonosító. Hibák jelzése.
6. Eljárások. Példa: `square`. Argumentum és eredmény. Eljárás feje és törzse. Argumentumminta, argumentumváltozó. Eljárásdeklaráció. Eljárás típusa. Fejkomment.
7. Relációk és feltételes kifejezések. `bool`, `false`, `true`.
8. Lokális deklarációt használó kifejezés (`let`-kifejezés).

## 2. előadás, okt. 4.

1. Példa lokális deklarációt használó kifejezés használatára: `xa8adikon`.
2. Ennesek (párok, hármások stb.). Ennes  $i$ -edik tagja; projekció. Nullas: `( )`; `unit`.
3. Ennesminta. `swap, val (x, y) = (y, x). min, max`.
4. Típusváltozó, polimorfizmus. Operátorok többszörös terhelése. Alapműveletek és relációk alapértelmezett típusa.
5. Rekurzív eljárások. Példa: `power`. Rekurzív eljárás alkalmazása, végrehajtása. Rekurzív hívás, leállási feltétel.
6. Eljárásdeklaráció mintaillesztéssel. Állandó és azonosító mint `minta`.
7. Egészosztás: `div, mod, quot, rem`.

**Gyakorló házi feladatok.** Írjon három eljárást `min3`, `mid3` és `max3` néven, amelyek három egész érték közül kiválasztják a legkisebbet, a középsőt, ill. a legnagyobbat.

**Haladóbbaknak.** Írjon olyan változatokat `min3c`, `mid3c` és `max3c` néven az előző eljárásokból, amelyek argumentumként kapják meg a két érték összehasonlítására használható, `int*int -> bool` típusú műveletet.

**Az előadásokon nem volt szó** az alábbi témákról, amelyekről a hallgatók más tárgyakban már tanultak; SML-beli megvalósításukat önállóan kell feldolgozniuk:

1. Fixpontos és lebegőpontos számok, `int` és `real` típus, `Int` és `Real` könyvtár.
2. Aritmetikai pontosság, kerekítési hiba. Négyzetgyökvonás Newton-módszerrel.

## 3. előadás, okt. 10.

1. Szintaxis. Szavak: állandók, operátorok, kulcsszók, azonosítók. Mondatok: kifejezések, deklarációk, minták, típusok, programok. Mondatok egy- és kétdimenziós ábrázolása. Zárójelezés. Szabad és kötött azonosítók, környezetek, nyitott és zárt mondatok. Eljárások ábrázolása hármasként.
2. Szemantika. Szemantikai helyesség, típuszabályok. Kifejezések, eljáráshívások, deklarációk és programok végrehajtása. Értelmezőprogramok feldolgozási lépései: lexikai elemzés, szintaktikai elemzés, szemantikai elemzés, végrehajtás. Szemantikai egyenlőség.
3. Névtelen eljárások ( $f_n$ - vagy lambda-jelölés). Kaszkádosított (angolul: *curried*), másnéven *részlegesen alkalmazható* eljárások. Ennessel paraméterezett (angolul: *uncurried*) eljárások.

4. Magasabbrendű eljárások. Példák: `sum`, `gauss`, `iter`, `sqrt`, `first`.
5. Eljárások és folyamatok: rekurzív eljárásból rekurzív folyamat, rekurzív eljárásból iteratív folyamat, iteratív eljárásból rekurzív folyamat.

#### 4. előadás, okt. 11.

1. Polimorf és monomorf eljárások. Típusémák és típusok. Típusémák példányosítása. `iter` polimorf változata. `iter` egy alkalmazása: `power`. Monomorf és polimorf deklarációk. Az `id` függvény. Monomorf és polimorf azonosítók. Nem egyértelmű deklarációk, szabad típusváltozók, érték-polimorfizmus. Polimorf eljárások specifikálása.
2. Újra a típuslevezetésről és az operátorok többszörös terheléséről.
3. Egyenlőségvizsgálatot megengedő, ún. egyenlőségi típusok és típusváltozók.
4. Azonosítók definiáló és alkalmazó előfordulása. Azonosítók kötött és szabad előfordulása. Lexikális kötés. Kötött azonosítók szisztematikus átnevezése. Egy mondat „megtisztítása”. Szemantikai elemzés, statikus kötés, monomorf és polimorf kötés. Végrehajtás, dinamikus kötés.

5

5. Az `fn`-jelölés kifejezőereje. Klózok. Szintaktikai édesítők: `e1 and also e2`, `e1 or else e2`, `if e1 then e2 else e3`, `op+`, `op*`, `op div`, `op<`, `op>=`, `op=` stb.

**Önálló feldolgozásra.** Tesztelőeljárás egy szám prím voltának eldöntésére.

#### 5. előadás, okt. 17.

1. Lista: rekurzív lineáris adatszerkezet, egy értékből és egy listából álló pár.
2. Üres lista: `nil`, `[]` Lista elemei. Lista feje, farka.
3. Listajelölések. A `cons` művelet: `::`. Listák összefűzése: `@`. Műveletek kötése, precedenciája.
4. Lista típusa. Polimorf lista.
5. `length`, `append`, `@`, `naív rev`, `revAppend`, `rev`, `concat`, `tabulate`.

6

## 6. előadás, okt. 18.

1. `map`, `filter`, `exists`, `all`. Típusuk levezetése.
2. `fold f e n` és `fold' f e n` származtatása `sum f n`-ből és `prod f n`-ből. Típusuk levezetése.
3. `fold` és `fold'` kiértékelése `fold f e 3` és `fold' f e 3` példáján.
4. `exp n k`. `iter f n` származtatása `exp n k`-ből; típusa.
5. `iter f n` és `fold' f e n` összefüggése.

`fold f e n`, `fold' f e n` és `iter f n` származtatását és néhány alkalmazását példákon keresztül mutatjuk be a <http://dp.iit.bme.hu/dp05a/dp05a-mrelj.sml> címről letölthető programfájlban.

**Önálló feldolgozásra.** `fold (op*) 1 4` és `fold' (op*) 1 4`, továbbá `fold (op-) 0 3` és `fold' (op-) 0 3` kiértékelésének összehasonlítása. Mutassa be a jobbrekurzív és a nemjobbrekurzív változatok kiértékelése közötti különbségeket!

## 7. előadás, okt. 24.

1. Redukciós eljárások listákra: `foldl`, `foldr`.
2. `plus`, `rev`, `length`, `append`, `revApp`, `concat`, `map`, `filter` megvalósítása `foldl`-lel, ill. `foldr`-rel.
3. Törzstényezőkre bontás: `primefac`.
4. További eljárások listákra: `hd`, `tl`, `nth`, `null`. Kivétel jelzése `raise`-zel.
5. Több klózból álló eljárások. Mintaillesztés. Diszjunkt minták. Átfedő minták. Nem minden esetet lefedő minták. Klózok, amelyek végrehajtására soha nem kerül sor.
6. `case`-kifejezés.
7. Több klózból álló kaszkádosított eljárások.

## 8. előadás, okt. 25.

1. Listák rendezése beszúrással: `insert`, `isort`. Polimorf változata: `isort`.
2. Az `order` típus. Rendezés fordított sorrendben: az `invert` eljárás.
3. Lexikális rendezések. A `lex` eljárás.
4. Rendezés összefuttással: `split`, `merge`, `msort`.

**Önálló feldolgozásra.** `partition` és `quicksort` definiálása.

## 9. előadás, nov. 5.

1. Adatkonstruktor ( állandó, függvény), típuskonstruktor ( állandó, függvény).
2. `datatype`-deklaráció.
3. Adatkonstruktor mint minta.
4. Felsorolásos (enumerációs) típus `datatype`-deklarációval.

5. Típusszinoníma `type` definícióval.
6. Kivétel deklarációja, kiváltása, kezelése: `exception`, `raise`, `handle`.
7. Szekvenciális kifejezés `;`, ill. `before` operátorral.
8. Példa:`testDouble` eljárás egyforma értékű listaelemek előfordulásának jelzésére.