ELTE IK - Programtervező Informatikus BSc

Záróvizsga tételek

22. Funkcionális programozás

TODO: Erősen hiányos!

Funkcionális programozás

Funkcionális programozási nyelvek jellemzői: lusta és mohó kiértékelési stratégiák jellemzése és összehasonlítása, hivatkozási helyfüggetlenség, statikus típusozottság, Curry-féle elv, margó szabály, alaptípusok, konverziók, függvények definiálása és típusozása, mintaillesztés, őrfeltételek, esetszétválasztás, rekurzió, lokális definíciók, magasabbrendű függvények, névtelen függvények, függvénykompozíció, halmazkifejezések, típusosztályok, parametrikus (paraméteres) és ad-hoc polimorfizmus, típusszinonimák, algebrai adattípusok definiálása.

1 Lusta és mohó kiértékelés

A funkcionális nyelvek kifejezés kiértékelése lehet lusta (lazy), vagy mohó (strict). A lusta kiértékelés során a függvények argumentumai csak akkor értékelődnek ki, ha azokra feltétlenül szükség van. A Clean ilyen lusta kiértékelést alkalmaz, míg az Erlang inkább a szigorú nyelvek közé tartozik. A mohó kiértékelés minden esetben kiértékeli a kifejezéseket, mégpedig olyan hamar, ahogyan az lehetséges. A lusta kiértékelés a inc 4+1 kifejezést elsőként (4+1)+1 kifejezésként interpretálná, míg a mohó rendszer azonnal 5 + 1 formára hozná.

2 Hivatkozási helyfüggetlenség

A kifejezések értéke nem függ attól, hogy a program szövegében hol fordulnak elő, vagyis bárhol helyezzük el ugyanazt a kifejezést, az eredménye ugyanaz marad. Ez a tisztán funkcionális nyelvekre igaz leginkább, ahol a függvényeknek nincs mellékhatása, így a kiértékelésük során sem változtatják meg az adott kifejezést.

3 Statikus típusrendszer

A statikus típusrendszerekben nem kötelező a deklarációk megadása, de követelmény, hogy a kifejezés legáltalánosabb típusát a fordítóprogram minden esetben ki tudja következtetni. A statikus rendszer alapja a Hindley-Milner polimorfikus típusrendszer.

Természetesen a típusok megadásának az elhagyhatósága nem azt jelenti, hogy azok nincsenek jelen az adott nyelvben. Igenis vannak típusok, csak a kezelésükről, valamint a kifejezések helyes kiértékeléséről a típuslevezető rendszer és a fordítóprogram gondoskodik.

4 Curry-féle elv

A Curry módszer a részleges függvény alkalmazás, ami azt jelenti, hogy a több paraméteres függvények visszatérési értéke lehet egy függvény és a maradék paraméterek. Ez alapján minden függvény tekinthető egyváltozós

függvénynek. Amennyiben a függvénynek két változója van, akkor csak az egyiket tekintjük a függvényhez tartozó paraméternek. A paraméter megadása egy új egyváltozós függvényt hoz létre, amit alkalmazhatunk a második változóra. A módszer több változó esetén is működőképes. Erlangban és F#-ban nincs ilyen nyelvi elem, de a lehetősége adott. Készítéséhez függvény kifejezést használhatunk.

5 Margó szabály

Összetartozó kifejezések csoportjának azonosítására és deklarációk hatáskörének korlátozására alkalmas a bal oldali margó szélességének változtatása.

6 Alaptípusok

TODO

7 Mintaillesztés

A mintaillesztés függvényekre alkalmazva azt jelenti, hogy a függvény több ággal rendelkezhet, és az egyes ágakhoz (clause) a formális paraméterek különböző mintáit rendelhetjük hozzá. A függvény hívásakor mindig az az ág fut le, amelynek a formális paramétereire a híváskor megadott aktuális paraméterek illeszkednek. Az OO nyelvekben az ilyen függvényeket overload függvénynek nevezzük. A mintaillesztés alkalmazható változók, és listák mintaillesztése esetén is, valamint elágazások ágainak (branch) a kiválasztására.

8 Magasabbrendű függvények

Funkcionális nyelvekben lambda-kifejezéseket, vagyis speciálisan leírt függvényeket adhatunk át más függvények paraméter listájában. Természetesen nem a függvényhívás kerül a paraméterek közé, hanem a függvény prototípusa.

Számos programozási nyelv lehetőséget ad arra, hogy változókba kössük a lambda-függvényeket, majd az így megkonstruált változókat később függvényként használjuk. A lambda-kifejezéseket elhelyezhetjük lista-kifejezésekben is. A függvénnyel való paraméterezés segítségünkre van abban, hogy teljesen általános függvényeket, vagy programokat hozzunk létre, ami kifejezetten hasznos szerver alkalmazások készítésekor.

9 Halmazkifejezések

A különböző nyelvi konstrukciók mellett a funkcionális nyelvekben számos különleges adattípust is találunk, mint a rendezett n-es, vagy ismertebb nevén a tuple, valamint a halmazkifejezések, és az ezekhez konstruálható lista generátorok. A lista adattípus a Zermelo-Fraenkel féle halmazkifejezésen alapul. Tartalmaz egy generátort, amely az elemek listába tartozásának feltételét írja le, valamint azt, hogy a lista egyes elemeit hogyan, és milyen számban kell előállítani. Ezzel a technológiával elvben végtelen listát is képesek vagyunk leírni az adott programozási nyelven, mivel nem a lista elemeit, hanem a lista definícióját írjuk le.