

Razvoj programskog jezika Miranda

Seminarski rad u okviru kursa
Dizajn programskih jezika
Matematički fakultet

Jovan Randelović
mr16211@alas.matf.bg.ac.rs

10. decembar 2019

Sažetak

U ovom radu su predstavljene osnovne karakteristike programskog jezika Miranda. Ukratko su opisani i jezici koji su uticali na njegov razvoj: Hope, SASL, ML i KRC. Dat je i prikaz razvojnog stabla koje obuhvata sve navedene jezike.

Sadržaj

1	Uvod	2
2	Osnovno o Mirandi	2
3	Razvojno stablo	3
3.1	Hope	4
3.2	SASL	4
3.3	ML	4
3.4	KRC	5
4	Zaključak	5
	Literatura	5

1 Uvod

U proučavanju programskih paradigmi funkcionalna paradigma igra značajnu ulogu. Program se u funkcionalnom programiranju tretira kao evaluacija matematičkih funkcija koja pritom izbegava stanja i promenljive podatke. Glavni akcenat je na primeni funkcija, za razliku od, na primer, imperativne paradigme, kod koje je naglasak na promeni stanja. Funkcionalno programiranje^[3] se zasniva na sledećim aktivnostima:

- Definisane funkcije
- Primena funkcije
- Kompozicija funkcija

Teorijska osnova funkcionalne paradigme leži u lambda računu, formalnom sistemu razvijenom tokom 1930-ih.

Funkcionalni programski jezici se više koriste na univerzitetima nego u komercijalnom razvoju softvera. Jedan od najpoznatijih i najuticajnijih programskih jezika koji se koristi i u komercijalne svrhe u okviru ove paradigme je Haskell, a jedan od jezika koji je najviše uticao na njega je programski jezik Miranda. Logo jezika Miranda prikazan je na slici 1.



Slika 1: Logo programskog jezika Miranda

2 Osnovno o Mirandi

Miranda¹ je čist funkcionalni programski jezik koji podržava polimorfizam i funkcije višeg reda. Kreiran je u periodu 1983-1986 od strane britanskog informatičara Davida Turnera sa Univerziteta u Kentu. Osnovni cilj je bio da se napravi moderan funkcionalni programski jezik, ugrađen u pogodno programersko okruženje, koji bi se koristio i za nastavu a i kao sredstvo opšte namene programiranja. Jezik je bio naširoko prihvaćen i imao je veliki uticaj na dalje razvijanje funkcionalne paradigme.

Sintaksa jezika je vrlo prijatna i sažeta. Program napisan u Mirandi je u proseku 5 do 15 puta kraći nego onaj napisan u Javi ili u C-u. Najčešće korišćena struktura podataka je lista, na primer:

```
1000 radni_dani = ["Pon", "Uto", "Sre", "Cet", "Pet"]
```

Moguće je proširivanje liste operatorom "++":

```
1000 dani = radni_dani ++ ["Sub", "Ned"]
```

¹Jezik je dobio ime po latinskoj reči *miranda*^[1] koja u prevodu znači "čuditi se", "zamisliti se".

Ostali korisni operatori za liste su ":" , koji umeće element na početak liste, "#" koji vraća dužinu liste , "!" koji služi za indeksiranje i operator "--" koji vrši oduzimanje.

Za liste čiji elementi čine aritmetički niz moguće je koristiti ".." , na primer:

```
1000 fac n = product [1..n]
```

Moguća je i specijalna iteracija nad listama. U sledećem primeru je prikazana lista koja sadrži kvadrate brojeva od 1 do 100:

```
1000 [ n*n | n <- [1..100] ]
```

Rešenje za poznat problem o raspoređivanju n dama na šahovskoj tabli tako da se nikoje 2 "ne napadaju" se može vrlo lako realizovati u Mirandi:

```
1000 queens 0 = [ [ ] ]
1002 queens (n+1) = [ q:b | b <- queens n; q <- [0..7]; safe q b ]
safe q b = and [ ~checks q b i | i <- [0..\#b-1] ]
checks q b i = q=b!i \\/ abs(q - b!i)=i+1
```

Elementi liste moraju biti istog tipa, ukoliko imamo elemente različitog tipa onda je reč o torci.

Miranda je potpuni jezik višeg reda[2] - funkcije se mogu prosleđivati i kao parametri i vraćati kao rezultat.

Primena funkcije je levo-asocijativna, kada napišemo " $f x y$ " to se parsira kao " $(f x) y$ ", što znači da je rezultat primene f na x funkcija koja se dalje primenjuje na y . Postoji prilično velika biblioteka standardnih funkcija.

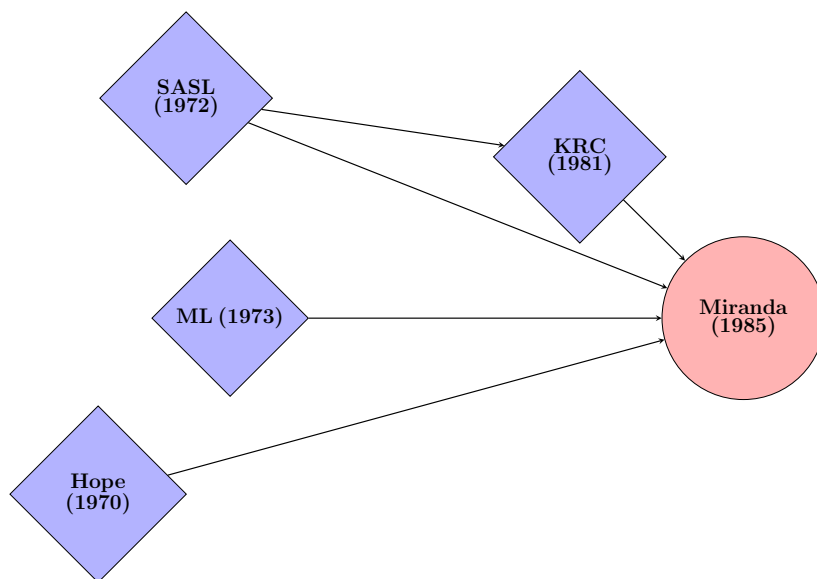
Jedna od karakteristika ovog jezika je i da podržava lenju evaluaciju, odnosno izbegava nepotrebna izračunavanja.

Miranda je strogo tipizirana i pored tri primitivna tipa (num, bool i char) moguće je i kreiranje korisnički definisanih tipova. Svi tipovi mogu biti polimorfni.

Miranda sistem je bio temeljno testiran i u svoje vreme je radio na 600 lokacija, uključujući 250 univerziteta.

3 Razvojno stablo

Na nastanak jezika Miranda uticali su jezici **Hope**, **SASL**, **ML** i **KRC**. Od jezika koji su nastali pod uticajem Mirande najviše se izdvaja **Haskell**. Razvojno stablo jezika je prikazano na slici 2.



Slika 2: Razvojno stablo jezika Miranda

3.1 Hope

Hope[8] je mali funkcionalni programski jezik razvijen tokom 1970-ih na Edinburškom univerzitetu. Neke od osnovnih karakteristike jezika su lenja evaluacija, polimorfno tipiziranje i funkcije višeg reda. Jednim delom je uticao na nastanak Mirande i **Haskella**. Hope je izveden iz jezika **NPL**, jednostavnog funkcionalnog jezika kog su razvili Rod Burstall i John Darlington u jednom od svojih radova. NPL i Hope su poznati po tome što su prvi jezici sa algebarskim² tipovima podataka. Miranda je od ovog jezika preuzela lenjo izračunavanje, polimorfno tipiziranje kao i kreiranje korisnički definisanih tipova.

3.2 SASL

SASL (St Andrews Static Language) je čist funkcionalni programski jezik koji je razvio David Turner na Univerzitetu St Andrews 1972, inspirisan jezikom **ISWIM**. Par godina kasnije, Turner ga je redizajnirao i u toj novoj formi bio je temelj za izgradnju jezika **KRC** i Miranda, s tim što SASL nije tipiziran dok Miranda ima polimorfne tipove. Sam Turner je u jednom od svojih radova[11] rekao da je Miranda u suštini SASL plus algebarski i polimorfni tipovi.

3.3 ML

ML (Meta Language) je funkcionalni programski jezik opšte namene. Kreirao ga je Robin Milner tokom ranih 1970-ih. Sintaksa je inspirisana uglavnom jezikom ISWIM, mada vuče korene i iz **Lispa** i često je bio okarakterisan kao "Lisp sa tipovima". Neke od karakteristika ovog jezika su funkcije prvod reda, automatsko upravljanje memorijom pomoću

²Algebarski tip podataka je vrsta kompozitnog tipa koji u sebi kombinuje više tipova.

sakupljača otpadaka, parametarski polimorfizam[9], statičko tipiziranje, algebarski tipovi podataka i rukovanje izuzecima. Neki smatraju da ML nije u potpunosti funkcionalan jezik jer dozvoljava promene stanja.

Jedan je od retkih jezika koji je u potpunosti preciziran i verifikovan koristeći formalnu semantiku. Primenjuje se u dizajnu programskih jezika kao i u oblastima kao što su kompilacija, dokazivanje teorema, bioinformatika i finansijski sistemi. Istorijski gledano, ML je bio zamišljen da razvije taktike dokaza u LCF dokazivaču teorema. Upotreba * ** ***...[11] za korišćenje polimorfih tipova u Mirandi vuče korene iz jezika ML.

3.4 KRC

KRC (**Kent Recursive Calculator**) je lenji funkcionalni programski jezik nastao u periodu od 1979 do 1981, zasnovan na jeziku SASL, sa ugrađenim mehanizmima za poklapanje obrazaca, logičkim uslovima i ZF izrazima. Miranda je od jezika KRC nasledila rad sa ZF izrazima[11] (pogledati odeljak 2 u ovom radu, primer 4, lista koja sadrži kvadrate brojeva od 1 do 100). Najviše se koristio za nastavu funkcionalnog programiranja na Univerzitetu u Kentu.

4 Zaključak

U ovom tekstu su ukratko iznete osnove razvoja programskog jezika Miranda. Predstavljeni su jezici koji su najviše uticali na njegov nastanak i razvoj kao i njegovo razvojno stablo. Za dodatne informacije o jeziku najbolje je koristiti zvaničnu stranu programskog jezika Miranda.

Literatura

- [1] Zvanična strana programskog jezika Miranda: <http://miranda.org.uk/>
- [2] D.A.Turner, *An Overview of Miranda*, December 1986
- [3] http://www.programskijezici.matf.bg.ac.rs/ppR/2019/predavanja/fp/funkcionalno_programiranje_slajdovi.pdf
- [4] Chris Clack, Colin Myers Ellen Poon, *Programming with Miranda*, Prentice Hall 1995
- [5] Simon Thompson, *Miranda: the Craft of Functional Programming*, Addison-Wesley 1995
- [6] Zvanična strana programskog jezika KRC: <http://www.krc-lang.org/>
- [7] https://en.wikipedia.org/wiki/Kent_Recursive_Calculator
- [8] [https://en.wikipedia.org/wiki/Hope_\(programming_language\)](https://en.wikipedia.org/wiki/Hope_(programming_language))
- [9] [https://en.wikipedia.org/wiki/ML_\(programming_language\)](https://en.wikipedia.org/wiki/ML_(programming_language))
- [10] [https://en.wikipedia.org/wiki/SASL_\(programming_language\)](https://en.wikipedia.org/wiki/SASL_(programming_language))
- [11] <https://www.cs.kent.ac.uk/people/staff/dat/tfp12/tfp12.pdf>