

Programski prevodioci 1

Tabele simbola

Tekući opseg važenja imena i pravila vidljivosti

- **Opseg važenja imena** je deo listinga programa u kojoj je neko ime vezano za određenu deklaraciju (a u vreme izvršavanja za određenu memorijsku lokaciju).
- Bilo koja linija koda sadrži se unutar jedne programske jedinice ili više programskih jedinica koje definišu opseg važenja imena.
- Većina savremenih jezika je **blokovski strukturirana** - dozvoljava da opseg važenja imena bude ugnežđen.

Opsezi važenja imena

- **Tekući opseg važenja** je opseg važenja imena definisan trenutnim najdubljim ugnježdenjem.
- **Otvoreni opsezi** su tekući opseg važenja i svi opsezi koji okružuju tekuću liniju koda.
- **Zatvoreni opsezi** su opsezi važenja imena definisani bilo kojom programskom jedinicom koja ne okružuje posmatranu liniju izvornog koda.
- Tekući opseg važenja nije fiksni već se menja dinamički zavisno od tačke posmatranja.

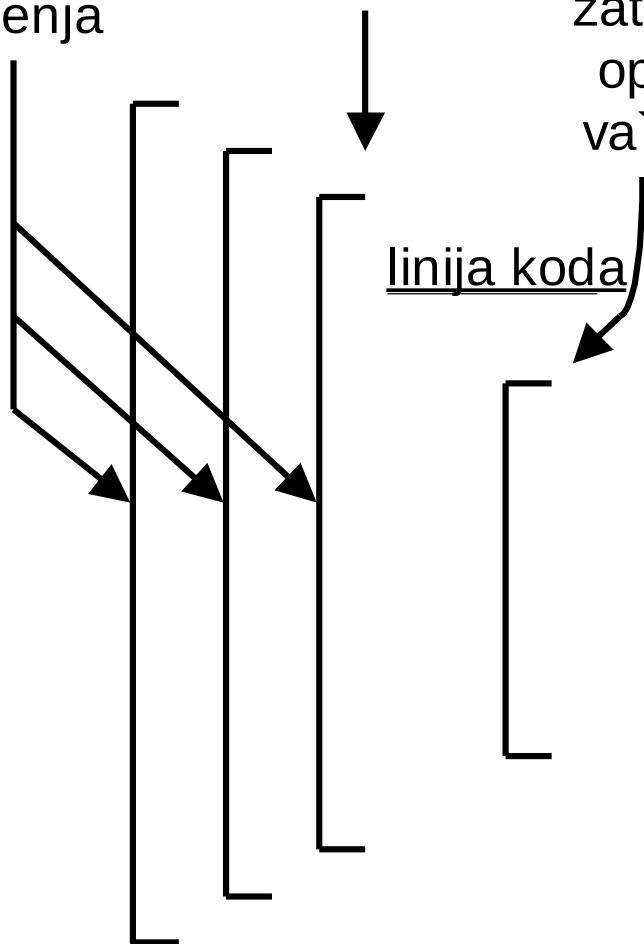
Opsezi važenja imena

otvoreni
opsezi
va`enja

teku}i opseg
va`enja

zatvoren
opseg
va`enja

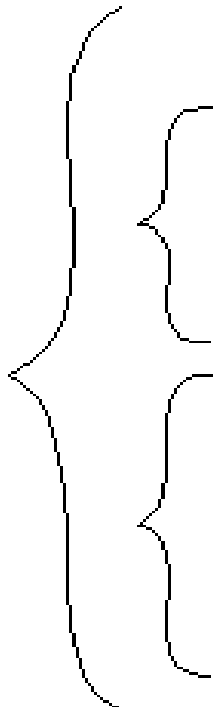
linija koda



Statička pravila vidljivosti imena

1. U bilo kojoj tački izvršnog programa pristupačna su samo imena deklarirana u tekućem opsegu važenja i u otvorenim opsezima važenja koji sadrže tekući opseg važenja
 2. Ako je ime deklarirano u više od jednog opsega važenja, za interpretaciju tog imena koristi se deklaracija čija je dubina najveća i najbliža tački obraćanja.
 3. Nove deklaracije se dozvoljavaju samo u tekućem opsegu važenja.
- Očigledna posledica ovih pravila je da pri zatvaranju nekog opsega, sve deklaracije zadate u njemu postaju nepristupačne.

Statička pravila vidljivosti - primer



```
declare  H,A,L:integer
begin
  declare  X,Y:real
  begin
    ...
    ...
  end
  declare  A,C,M:character
  begin
    ...
    <tekuca pozicija>
    ...
  end
end
```

- Na mestu tekuća pozicija u programu vidljiva su sledeća imena:
A,C,M:character
H,L,:integer
- Nisu vidljivi X i Y jer je njihov blok zatvoren, dok se vidi A iz treće deklaracije a ne iz prve zbog drugog pravila vidljivosti.

Dinamička pravila vidljivosti imena

- Za razliku od statičkih pravila vidljivosti, gde se uvidom u listing programa može otkriti veza imena sa deklaracijom, kod dinamičkih pravila vidljivosti tok izvršavanja programa određuje vezivanje upotrebe promenljive sa deklaracijom.
- Važi da se ime vezuje za najsvežiju aktivnu deklaraciju u vreme izvršavanja programa.
- Samo neki dijalekti Lispa, Perl imaju ovakva pravila, svi ostali programski jezici koriste statička pravila .

Primer za **dinamičku** vidljivosti imena

```
var a : integer;  
procedure first  
  a := 1;  
  
procedure second  
  var a : integer;  
  first();  
  
begin  
  a := 2;  
  second();  
  write_integer(a);  
end;
```

- Dodela a:=2 u glavnom programu odnosi se na globalno a; poziv second() aktivira lokalnu deklaraciju a u second; poziv first() i dodela a:=1 menja lokalno a; po povratku iz second deaktivira se lokalno a; write ispisuje globalno a, tj. **2**
- u slučaju statičke vidljivosti, a:=1 odnosi se uvek na globalno a, tako da se na kraju ispisuje **1**

Realizacija tabele simbola

- U zavisnosti od opsega važenja postoje dva pristupa realizaciji tabele simbola:
 1. pojedinačna tabela simbola za svaki opseg važenja
 2. jedinstvena tabela za sve opsege važenja

Pojedinačne tabele za svaki opseg važenja

- Kod pojedinačnih tabela simbola za svaki opseg važenja stek mehanizam je pogodan za realizaciju. Pri otvaranju novog opsega stavlja se tabela simbola na stek, a po zatvaranju opsega skida se sa steka. Umesto cele tabele može se koristiti i samo ukazatelj na tabelu.
- Detaljnije ćemo proučiti ovu varijantu kod tabele simbola za Mikrojavu.

Pojedinačne tabele za svaki opseg važenja

- Nedostaci ovog pristupa su:
 - i. Ako želimo da pristupimo nekom nelokalnom imenu moramo pristupati većem broju tabela simbola, pogotovo ako želimo da pristupimo nekom globalnom imenu.
 - ii. Ako su tabele realizovane kao heš tabele, za svaki opseg važenja se rezerviše memorijski prostor za tabelu simbola. Na duboko ugnježdenim nivoima gde ima malo imena gubitak prostora je najveći (nije veliki problem danas kada računari imaju puno memorije). Ovaj problem se ne javlja ako su tabele organizovane u obliku binarnog stabla.

Jedinstvena tabela za sve opsege

- Sva imena koja se pojave u toku prevodjenja se smeštaju u jedinstvenu tabelu simbola. Pošto se može desiti da u različitim opsezima važenja imamo ista imena, svakom imenu se pridruži opseg važenja za koji je to ime definisano. (Posebna globalna promenljiva „broji” opsege, inkrementira se pri otvaranju opsega).

Realizacija jedinstvene tabele u vidu heš tabele

- Primer

```
{ int M, A, L;
```

```
...
```

```
{float X, Y;
```

```
...
```

```
}
```

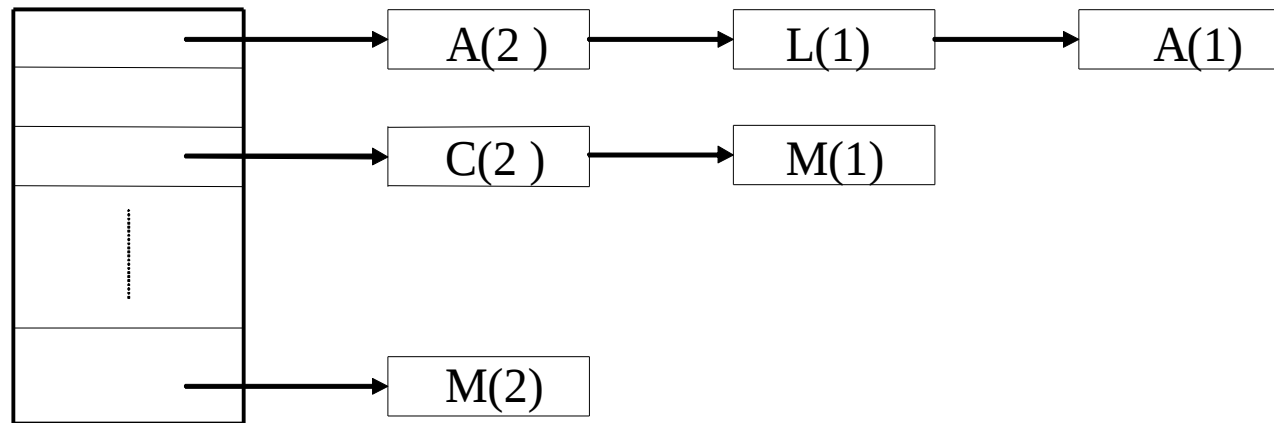
```
{char A, C, M;
```

```
...
```

```
//trenutna pozicija
```

```
}
```

```
}
```



heš tabela

Realizacija jedinstvene tabele u vidu heš tabele

- Prednosti heš tabele simbola su:
 - i. Brži je pristup simbolu jer se pretražuje samo jedna tabela, i obično su ti traženi simboli na početku lanaca.
 - ii. (manje važno) Memorijski prostor je efikasnije iskorišćen jer imamo samo jedno zaglavlje tabele, ali je taj dobitak umanjen zato što se uz svako ime dodaje prostor za opseg važenja.

Realizacija jedinstvene tabele simbola korišćenjem binarnog, leksikografski uređenog, stabla

Primer:

```
{ int M, A, L;
```

```
...
```

```
{float X, Y;
```

```
...
```

```
}
```

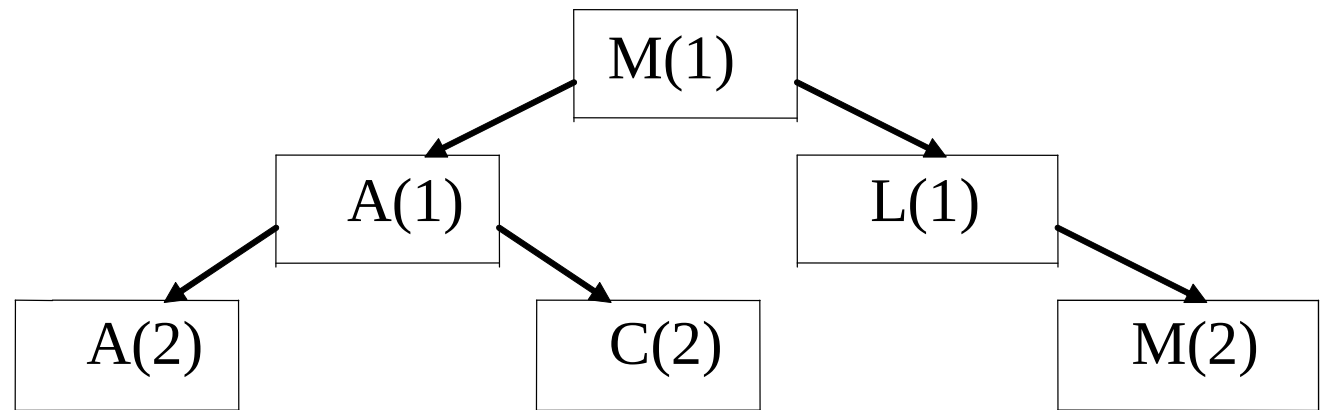
```
{char A, C, M;
```

```
...
```

```
// trenutna pozicija
```

```
}
```

```
}
```



Realizacija jedinstvene tabele simbola korišćenjem binarnog, leksikografski uređenog, stabla

- Kod ovog načina realizacije tabele simbola, svi simboli formiraju stablo. Ubacivanje novih simbola pri otvaranju novog opsega važenja i izbacivanje simbola pri zatvaranju opsega važenja zahteva pretraživanje stabla do listova, što nije baš efikasno pa se ne preporučuje ova struktura podataka ukoliko imamo jednu tabelu za sve opsege važenja. Binarno stablo se može koristiti kada imamo pojedinačne tabele za sve opsege važenja.

Realizacija jedinstvene tabele u vidu steka

- U ovom slučaju na vrhu steka se nalaze simboli iz tekućeg opsega važenja, i razdvojeni su od simbola iz prethodnog opsega važenja sa markerom. Kada se opseg zatvara sve što se nalazi na steku iznad markera se skida sa steka.
- Ovakva tabela je korišćena u ranim verzijama prevodilaca.

