SINTAKSA PROGRAMSKIH JEZIKA

Sintaksa, semantika i pragmatika jezika

- Da bi se program preveo na mašinski jezik neke platforme potrebno je:
  - defnisati šta su **ispravni programi** programskog jezika,
  - kao i defnisati koja izračunavanja odgovaraju naredbama programskog jezika.
- Pitanjima ispravnosti programa bavi se **sintaksa** programskih jezika (i njena podoblast leksika programskih jezika).
- Pitanjem značenja programa bavi **semantika** programskih jezika.
- Pitanjima izražajnosti bavi se **pragmatika** programskih jezika.

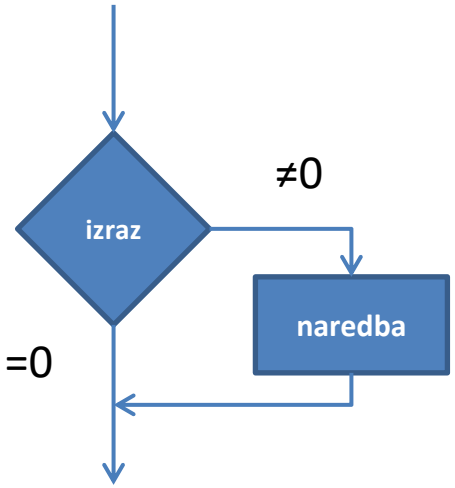
Sintaksa i semantika

- **Sintaksa jezika** - skup pravila na osnovu kojih se pišu konstrukcije jezika.
- Primenom sintaksnih pravila utvrđujemo da li je određena konstrukcija pravilna.
- *Sintaksne greške* su formalne (pogrešno otkucana reč, spojene dve reči,...) i prevodilac ih lako otkriva.

- **Semantika** određuje značenje pojedinih konstrukcija odnosno programa u celini.
- *Semantičke greške* su vezane za logiku programa i njih otkriva sam programer.

Sintaksa, semantika i pragmatika upravljačke strukture selekcije

1. if (izraz) naredba

2. 

3. if (x==1) {y=1;z=2;}

Sintaksa programskih jezika

- **Lingvistika prirodnih jezika** obuhvata u oblasti gramatike *fonetiku*, *morfologiju* (nauku o oblicima) i *sintaksu*, a pored toga *semantiku* i *leksikologiju*.
- U oblasti programskih jezika otpada potreba za fonetikom i morfologijom, pa se njihova **gramatika svodi jedino na sintaksu**.
- Da bi se definisala ili opisala sintaksa nekog jezlka potrebno je posedovati **specijalan jezik za opis sintakse**.
- Takav jezik, koji služl za opis nekog drugog jezika naziva se **metajezik**.
- Što se tiče semantike nekog jezika, ona se najčešće izražava **primenom prirodnih jezika**.
- **Prirodni jezici**, služe jedan drugome kao metajezici, jer se svaki jezik može opisivati i definisati sredstvima nekog drugog postojećeg prirodnog jezika.

Sintaksa programskih jezika

- Da bi se otklonile zabune prilikom definicije sintakse programskih jezika, važno je da se metajezik u potpunosti **razlikuje od jezika kojeg opisuje**.
- U oblasti sintakse programskih jezika primenjuje se **nekoliko metajezika**.
- Metajezici **ne podležu** nikakvim međunarodno usvojenim **standardima**, tako da postoji dosta varijacija u načinu označavanja i korišćenja.

Jezici za opis sintakse

- Sintaksa programskog jezika opisuju se pomoću posebnog jezika - metajezika.
  - Bekus-Naurova forma (BNF)
  - Proširena Bekus-Naurova forma (EBNF)
  - Sintaksni dijagrami
  - Sintaksna notacija sa zagradama

Sintaksa

- **Neterminali**: metaizrazi koji se moraju dalje objašnjavati (izraz, naredba)
- **Terminali**: metaizrazi koji se ne moraju dalje objašnjavati (if, while, tokeni)

BNF notacija

- BNF notaclja je metajezik razvijen 1960. godine prilikom definicije programskog jezika ALGOL 60.
- ALGOL 60 potiče od međunarodnog komiteta kome su bitne doprinose dall J. W. Backus i P. Naur, pa BNF predstavlja skraćenicu od "Backus-ova normalna forma" ili od "Backus-Naur-ova forma".
- Pri tome može biti interesantna činjenica da se kasnije ustanovilo da se srodna notacija koristila još u antičko doba za opis Sanskrita.

BNF notacija - metasimbol zagrada < >

- Prvi osnovni element BNF notacije je metasimbol zagrada < > koje služe za definiciju svih složenih objekata koji se javljaju u posmatranom programskom jeziku (**neterminala**).
- Tako će <x> označavati "objekt zvani x".
- Ako definišemo prirodan broj on će biti označen sa <prirodan broj>.
- Na sličan način, da bi definisali sintaksu nekog programskog jezika biće nam potrebne precizne definicije objekata kao što su:
  - <aritmetički izraz>,
  - <instrukcija IF>,
  - <instrukcija WHILE>, i mnogi drugi.
- **Terminali** se prikazuju bez ikakvih dodatnih znakova.

BNF notacija - metasimbol vertikalna crta (I)

- Drugi osnovni element BNF notacije je vertikalna crta (I) koja označava ekskluzivnu disjunkciju,na primer:

  <p> | <q>

  označava da na tom mestu postoje samo dve mogućnosti: ili će se primeniti objekt <p>, ili će se primenitiobjekt <q>.
- Ništa drugo nije dozvoljeno.

BNF notacija - metasimbol dodele vrednosti (::=)

- Treći osnovni element BNF notacije jeste metasimbol dodele vrednosti (::=).
- Ovaj simbol dodele vrednosti primenjuje se u metajeziku zato jer se znak = koristi u programskom jeziku kao relacioni operator, a znak := kao znak dodele vrednosti, pa je bilo neophodno uvesti neki poseban znak za dodelu vrednosti u metajeziku i tu je usvojen ::= .
- Na primer, definiciju

<x> ::= <p> | <q>

interpretiramo tako što tvrdimo da "objekt x može da bude bilo objekt <p>, bilo objekt <q>" (ali ništa drugo izuzev toga).

BNF notacija - lančanje, rekurzija

- Lančanje:

  <naredba WHILE>::=(<izraz>)<naredba>

- Rekurzija:

<identifikator>::=<slovo>|<identifikator><slovo>
|<identifikator><cifra>

BNF notacija - ponavljanje { }

- BNF notacija se često dopunjava vitlčastim zagradama koje označavaju ponavljanje obuhvaćenog objekta konačan ili beskonačan broj puta.
- U opštem slučaju oznakom:

$$\{\langle x \rangle\}_{m}^{n}$$

definišemo da se objekt **<x>** ponavlja (tj. lanča) od m do n puta.

- Ako se u ovoj oznaci izostavi m smatra se da je m=1, a ako se izostavi n smatra se da je n=+∞.
- Prema tome, ako se izostavi i m i n onda to znači ponavljanje od 1 do neograničeno puta, tako da su definicije:

<kardinalan broj> ::= {<cifra>}

i

<kardinalan broj> ::= <cifra> | <kardinalan broj> <cifra>

međusobno ekvivalentne.

EBNF notacija

- BNF notacija sa nizom proširenja i modifikacija naziva se proširena BNF notacija ili EBNF (Extended Backus-Naur Form).
- Ova notacija polazi od činjenice da se u većini realnih programskih jezika terminalni simboli navode pod navodnicama ili pod apostrofima.
- Ako se to usvoji i u metajeziku onda otpada potreba da se nazivi objekata, radi razlikovanja, pišu unutar zagrada tipa <>, već se mogu pisati direktno bez zagrada.
- Pored toga, otpada potreba da se za dodelu vrednosti u metajeziku primenjuje glomazni izraz ::= već se može direktno koristiti običan znak jednakosti (tj. "=" označava jednakost kao deo opisivanog jezika, dok se = koristi u metajeziku).
- Na ovaj način u znatnoj meri se pojednostavljuju neki od iskaza u metajeziku.

EBNF notacija

- Ako se za označavanje terminalnih simbola koriste znaci navoda, kao u sledećim primerima,

"[" , "]" , "begin" , "end" ,

onda treba imati način kako se u iskazima označava sam znak navoda kad se javi u ulozi terminalnog simbola.

- U tu svrhu se znak navoda navodi dva puta, tako da se sam znak navoda označava sa """" .
- Osnovne komponente EBNF notacije obuhvataju i one elemente koje smo već koristili u okviru BNF notaclje.

Metasimboli EBNF notacije i njihovo značenje

=	jednako po definiciji
|	ekskluzivno ili
.	obavezna oznaka kraja svakog sintaksnog pravila
Objekt	neterminalni simbol Objekt
"X"	terminalni simbol X
{Simbol}	terminalni ili neterminalni Simbol se ponavlja 0 ili više puta
[Simbol]	terminalni ili neterminalni Simbol se ponavlja 0 ili 1 put
(A | B)	zagrade za grupisanje u metalingvističkim izrazima (A ili B)

Sintaksni dijagrami

- Analitički zapisi sintakse mogu u slučaju komplikovanih metalingvističkih formula da budu dosta nepregiedni.
- Stoga se često pribegava zapisu sintakse u grafičkoj formi, jer se na taj način mogu dosta pregledno sagledati sve aiternative koje postoje u datoj jezičkoj konstrukciji.
- U tu svrhu se koriste specifični usmereni grafovi koji se nazivaju **sintaksni dijagrami**.

Sintaksni dijagrami - terminali

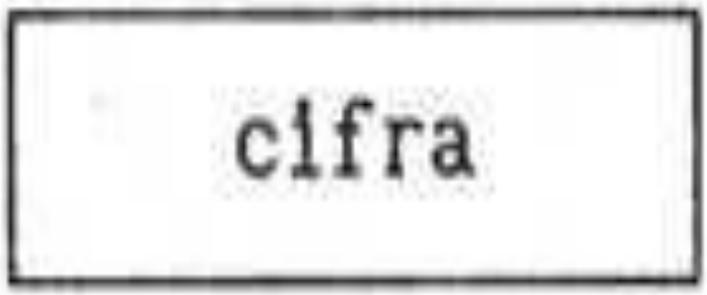
- TERMINALNI SIMBOLI koji predstavijaju elemente izvornog jezika pišu se onako kako se pojavljuju u izvornom jeziku i upisuju se unutar kružnih ili ovalnih simbola što ilustruju sledeći primeri:

Sintaksni dijagrami - poteg

- USMERENI POTEG (grana u grafu sintaksnog dijagrama) kojim se povezuju terminalni simboli i drugi objekti u okviru sintaksnih dijagrama predstavlja odsustvo bilo kakvih objekata, odnosno prazno:

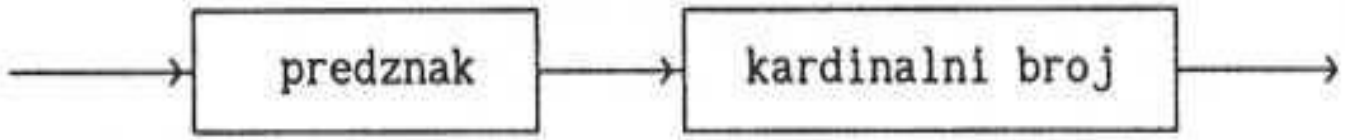
Sintaksni dijagrami - metalingvističke promenljive

- METALINGVISTIČKE PROMENLJIVE (objekti jezika) označavaju se malim slovima upisuju u pravougaone okvire. Na primer:



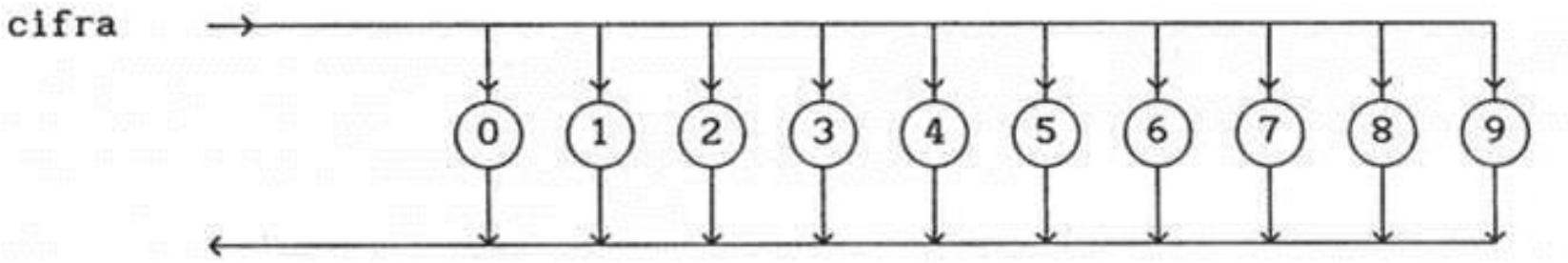
Sintaksni dijagrami - lančanje

- LANČANJE OBJEKATA prikazuje se u vidu niza simbola objekata povezanih potezima. Na primer na sledeći način definišemo objekt *ceo broj* koristeći prethodno definisane objekte *predznak* i *kardinalni broj*:



Sintaksni dijagrami - ekskluzivna disjunkcija

- EKSKLUZIVNA DISJUNKCIJA objekata predstavlja se skupom paralelnih grana grafa koje izlaze iz datog polaznog čvora i stiču se u zajedničkom završnom čvoru. Na primer, sledeći dijagram prikazuje definiciju objekta *cifra:*

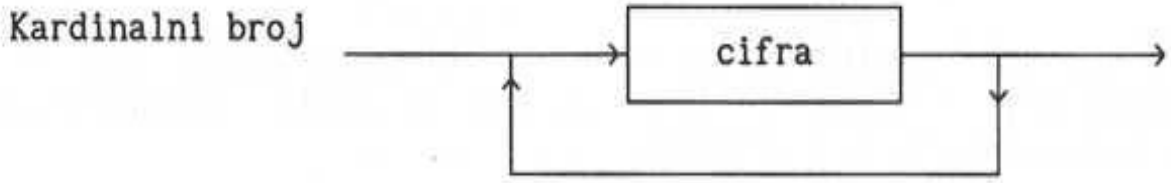


Sintaksni dijagram

- SINTAKSNI DIJAGRAM je usmereni graf koji ima jednu ulaznu tačku i jednu izlaznu tačku i služi za definiciju sintaksno ispravnih objekata i konstrukcija programskog jezika.
- Svaki sintaksno ispravan objekt definiše se kao jedna od putanja koje kroz sintaksni dijagram idu od ulaza do izlaza.

Sintaksni dijagrami - rekurzija

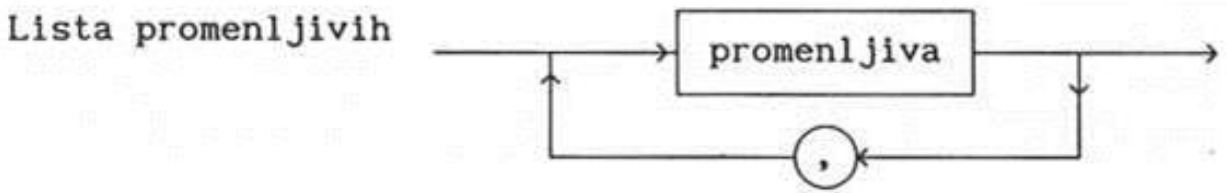
- Rekurzivne definicije svode se na pojavu povratne sprege u sintaksnim dijagramima. Na primer, sledeći dijagram pokazuje definiciju kardinalnog broja:



- Putanja koja vodi od ulaza do izlaza prolazi barem jednom kroz blok *cifra.*
- Zatim se možemo povratnom vezom vratiti na ulaz i proizvoljan broj puta proći kroz blok cifra pre nego izađemo; takva putanja koja povezuje ulaz i izlaz jasno definiše kardinalni broj kao niz od jedne ili više cifara.
- Ovakav sintaksni dijagram ne obuhvata ograničenje broja prolaza kroz pojedine grane. Stoga broj cifara u ovako definisanom kardinalnom broju nije ograničen.

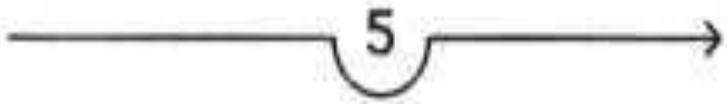
Sintaksni dijagrami - separatori

- Ako su elementi nekog niza razdvojeni separatorima onda se separatori pojavljuju u povratnoj vezi.
- Na primer, lista promenljivih razdvojenih zarezima definiše se na sledeći način:



Sintaksni dijagrami - ograničenje

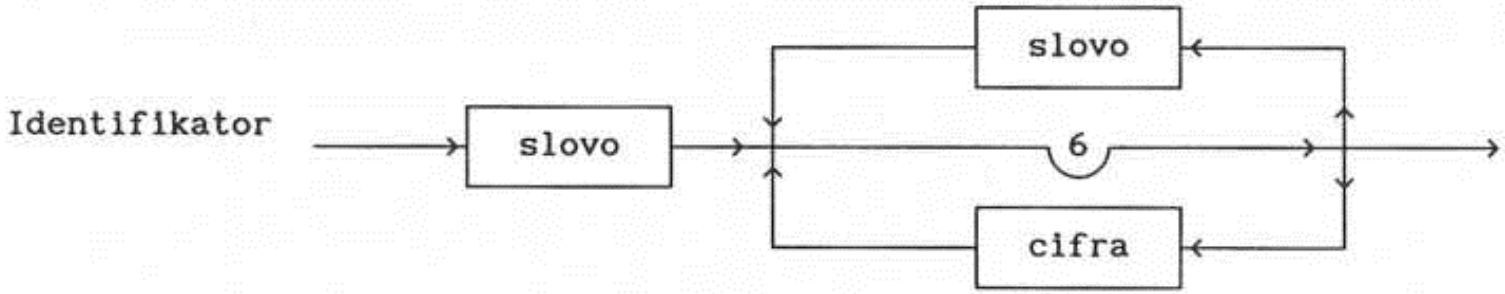
- Ponekad se broj prolaza kroz neku granu sintaksnog dijagrama može ograničiti.
- Takav poteg sa ograničenjem prikazuje se sa oznakom maksimalnog broja prolaza kroz navedenu granu.
- Na primer, kroz sledeću granu:



nije dozvoljeno proći više od 5 puta.

Sintaksni dijagrami - ograničenje

- Koristeći navedenu notaciju možemo identifikatore koji ne smeju biti duži od 6 znakova definisati na sledeći način:



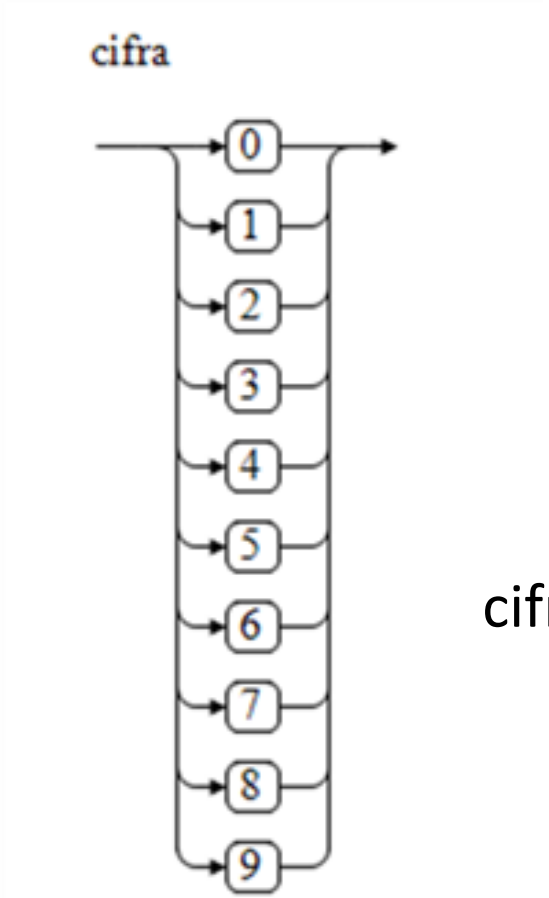
Tipovi podataka u C jeziku

- Podaci koji obrazuju neki tip podataka predstavljaju vrednosti ovoga tipa podataka.
- Tip podataka je, osim vrednostima koje ga obrazuju, određen i operacijama, definisanim za te vrednosti.

OZNAČAVANJE VREDNOSTI I OPERACIJA CELOG TIPA

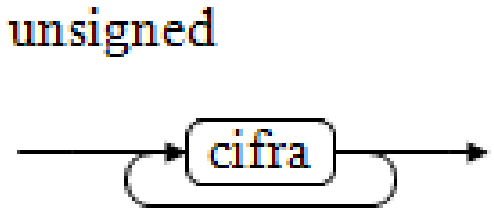
- Za označavanje nule i prvih devet prirodnih brojeva predviđene su posebne oznake koje se nazivaju cifre.
- Cifre su navedene u sintaksnom dijagramu, odnosno u sintagramu cifra, ali i u obliku proširene Bakus-Naurove forme (Extended Bacus Naur Form, EBNF),

Cifra

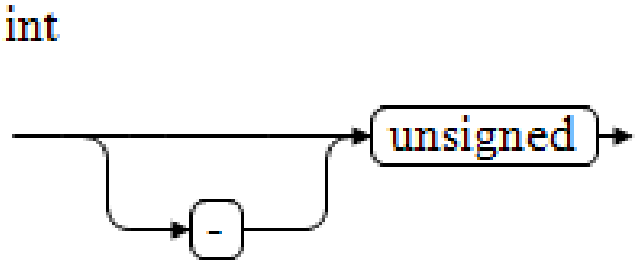


cifra=("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9").

Neoznačeni i označenu celi broj (unsigned i int)

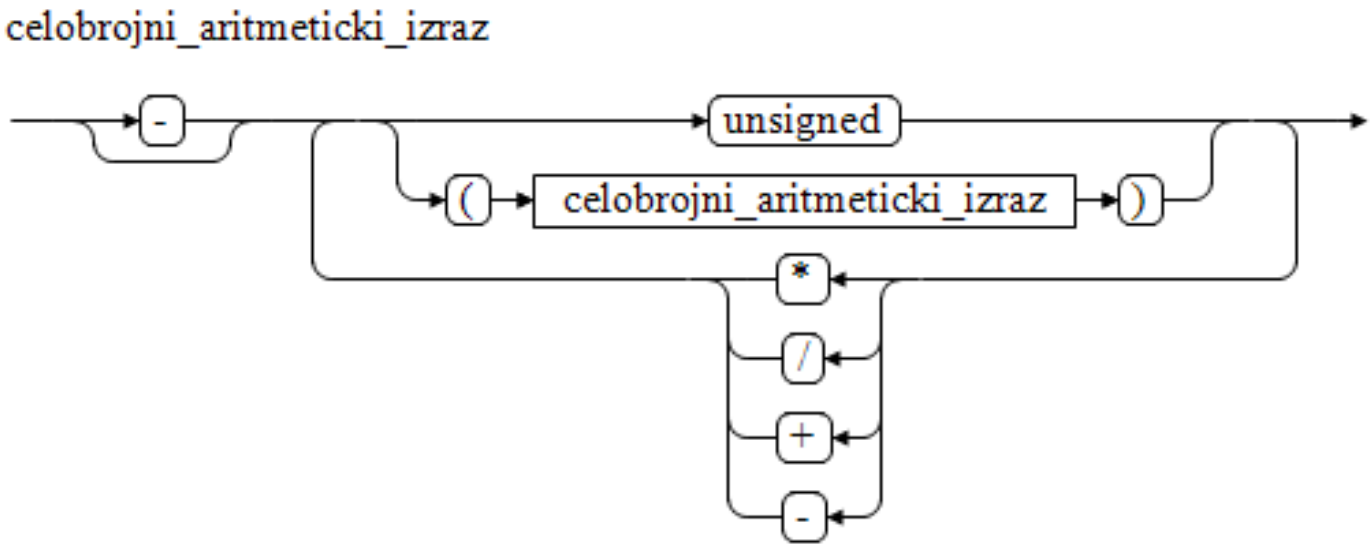


unsigned=cifra{cifra}.



int=["-"]unsigned.

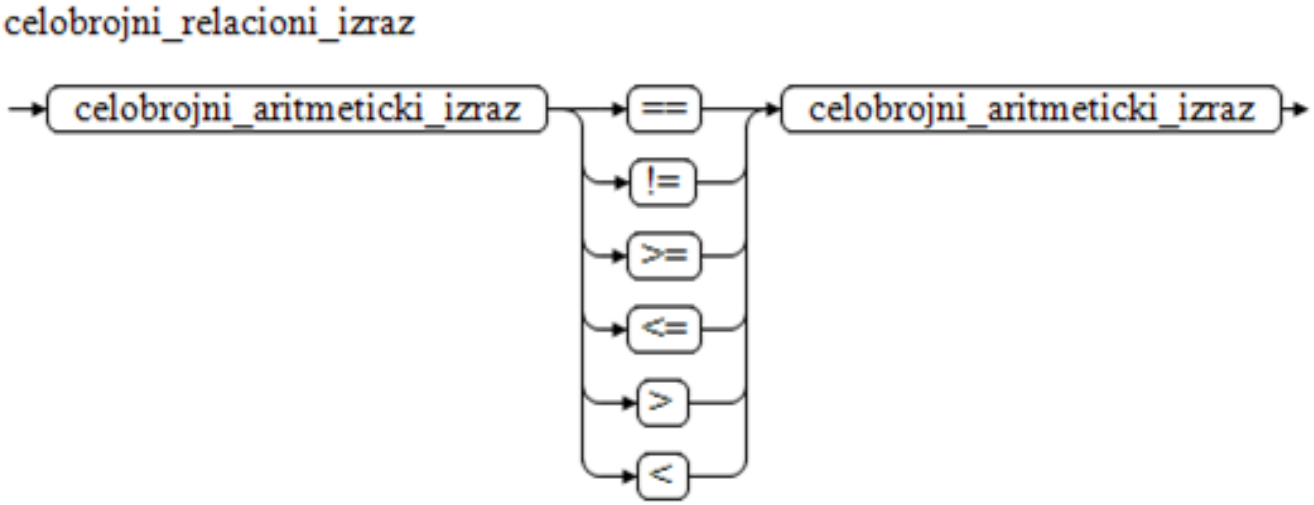
Celobrojni aritmetički izraz



celobrojni_aritmeticki_izraz=["-"](unsigned|"("celobrojni_aritmeticki_izraz")")

{("*"|"/"|"+"|"-")(unsigned|"("celobrojni_aritmeticki_izraz")")}.

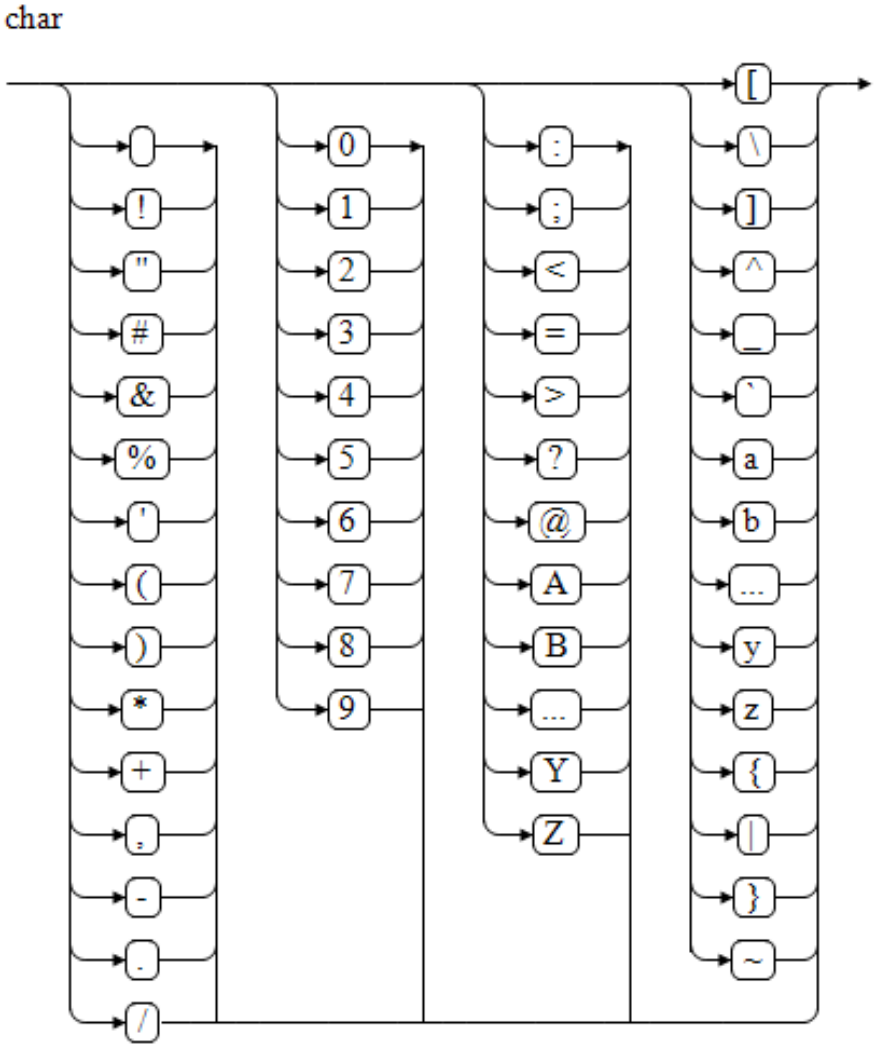
Celobrojni relacioni izraz



celobrojni_relacioni_izraz=celobrojni_aritmeticki_izraz

("=="|"!="|">="|"<="|">"|"<")celobrojni_aritmeticki_izraz.

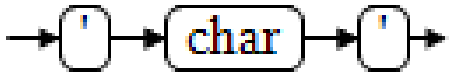
Karakter (char)



char=
(
" "|"!"|"'"|"#"|"&"|"%"|"&"|"'"|"("|")"|
"*"|"+"|","|"-"|"."|"/"
|"0"|"1"|".."|"8"|"9"
|":"|";"|"<"|"="|">"|"?"|"@"
|"A"|"B"|".."|"Y"|"Z"
|"["|"\"|"]"|"^"|"_"|"`"
|"a"|"b"|".."|"y"|"z"
|"{"|"|"|"}"|"~"
).

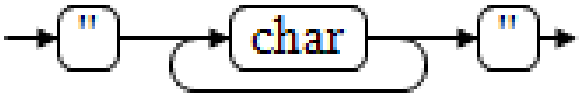
Znakovni izraz i string

znakovni_izraz



znakovni_izraz = """char""".

string



string = """char{char}""".