

Parsiranje odozdo nagore

- * Problem parsiranja sastoji se od nalaženja izvođenja određene rečenice pomoću date gramatike, ako takvo izvođenje postoji. U slučaju parsiranja odozdo nagore traži se **desno izvođenje**. Pokazaćemo kako se to može uraditi pomoću gramatike koja generiše jezik $x^+ y^+$.

$$S \rightarrow XY$$

$$X \rightarrow xX$$

$$X \rightarrow x$$

$$Y \rightarrow yY$$

$$Y \rightarrow y$$

IDE IDE
13
R-SPUST = AD 2,3
+ ovo

Tražićemo izvođenje rečenice $xxxxyy$:

$$S \rightarrow XY \rightarrow Xy \rightarrow Xyy \rightarrow xXyy \rightarrow xxXyy \rightarrow xxxxyy$$

- * Kod parsiranja odozdo nagore koraci izvođenja se ne identifikuju u tom redosledu, nego u suprotnom, od rečenice ka startnom simbolu, pa je desno izvođenje za to parsiranje:

$$xxxxyy \rightarrow xxXyy \rightarrow xXyy \rightarrow Xyy \rightarrow Xy \rightarrow XY \rightarrow S$$

- * Na svakom koraku primjeni se gramatičko pravilo tako što se njegova desna strana zamjeni lijevom, koja se sastoji od samo jednog simbola.

- * Međutim, ako rečenicu $xxxxyy$ čitamo slijeva nadesno, nije odmah očigledno zašto se prvo x zamjeni sa X u prvom koraku parsiranja - isto i za prvo y . Jasno je da je za parsiranje, osim samih pravila, potrebno još informacija.
- * U parsiranju odozdo nagore desne strane pravila se ne prepoznaju dok nisu u potpunosti očitane, te prema tome postoji potreba da se čuvaju djelimično prepoznate desne strane pravila, dok se ne zamijene odgovarajućim lijevim stranama. Stak predstavlja odgovarajuću strukturu za čuvanje tih djelimično prepoznatih nisti. (LIFO) Prema tome, detaljan opis procesa parsiranja odozdo nagore uključuje i prikaz sadržaja tog steka.
- * Koraci svakog od koraka parsera se sastoje od 2 tipa akcija:

1. stavbanje poslednjeg očitanoog simbola na stek - shift akcija (pomjeranje)
2. Zamjena niske koja je na vrhu steka primjenom gramatičkog pravila - reduce akcija (redukovanje).

Ulaзни string	stek	Pravilo	Rečenična forma	S/R
xxxxyy	x		xxxxyy	S
xxxyyy	xx		xxxyyy	S
xxxxyy	xxx		xxxxyy	S
xxxxyy	xxxX	$X \rightarrow x$	xxxxyy xxxyy	R
xxxxyy	xxX	$X \rightarrow xX$	xxxxyy xxXyy	R
xxxxyy	X	$X \rightarrow xX$	xxxxyy xXyy	R
xxxxyy	Xy		Xyy Xyy	S
xxxxyy	Xyy		Xyy Xyy	S
xxxxyy	XyyY	$Y \rightarrow y$	XyyY Xyy	R
xxxxyy	XY	$Y \rightarrow yY$	XY	R
xxxxyy	S	$S \rightarrow XY$	S	R

Neophodan uslov za reduce akciju je to da desna strana nekog pravila na vrhu steka, inače dolazi do shift akcije i sledeći simbol će biti očitao na vrh steka. Ipak, pojava desne strane nekog pravila na vrhu steka nije i dovoljan uslov da bi došlo do redukcije. Takođe je moguće identifikovati desnu stranu više od jednog pravila na vrhu steka, tako da može postojati više mogućih redukcija. Ako u određenoj fazi parsiranja izgleda moguća ili shift ili reduce akcija, onda se kaže da postoji shift-reduce konflikt. Ako izgledaju moguće 2 ili više redukcija, onda se radi o reduce-reduce konfliktu.

* Da bismo došli do determinističkog metoda parsiranja, mora da postoji način donošenja tih odluka. U praksi, to se rešava korišćenjem:

1. dosadašnjeg toka parsiranja
2. informacija vezanih za nailazeće simbole.

kao i kod parsiranja odzgo nadole.

Često se za donošenje tih odluka koristi jedan nailazeći simbol.

* Gramatika kod koje je moguće riješiti sve konflikte parsiranja odozdo nagore na osnovu fiksirane količine informacija koje se odnose na dosadašnji tok parsiranja i ograničene količine nailazećih simbola naziva se LR(k) gramatika (čitanje slijeva nadesno, desna izvođenja, k nailazećih simbola).