

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Blue D.

Seminarski rad iz predmeta
Prevođenje programskih jezika

Zadatak broj 3029
Zagreb, siječanj 2007.

Seminarski rad iz predmeta Prevođenje programskih jezika

Student: Blue D.

Matični broj studenta:

Zadatak broj 3029: Programski ostvariti parser Pomakni/Reduciraj. Ulaz u program je tablica parsera u slobodnom formatu. Izgraditi tablice parsera za nekoliko primjera gramatika i pokazati ispravnost rada ostvarenog parsera.

UVOD

PARSIRANJE OD DNA PREMA VRHU

Parsiranje od dna prema vrhu započinje gradnju generativnog stabla listovima. Listovi stabla označeni su završnim znakovima gramatike. Gradnja stabla nastavlja se primjenom desnih strana produkcija gramatike na prethodno izgrađene čvorove. Budući da se u svakom koraku smanjuje (reducira) ili ostaje isti broj znakova u međunizu, produkcije se u postupku parsiranja od dna prema vrhu nazivaju *redukcije*. Ako se izgradi generativno stablo koje ima korijen označen početnim nezavršnim znakom gramatike, niz w je u jeziku koji generira zadana gramatika. Znakovi niza w čitaju se slijedno slijeva nadesno. Na temelju pročitanih znakova izabere se odgovarajuća redukcija. Učinkovitost postupka parsiranja ovisi o preciznosti kojom je moguće odrediti koju je redukciju potrebno primijeniti u datom koraku parsiranja.

Na temelju znaka a stoga i pročitano a znaka iz niza w , parser primjenjuje jednu od četiri akcija:

- **Akcija Pomakni:**
Pročitani znak iz niza w stavlja se na vrh stoga, a pokazivač na nizu w se povećava za jedan, tj. pomiče se za jedan znak u desno.
- **Akcija Reduciraj:**
Ako je na vrhu stoga uzorak koji se treba zamijeniti, onda se primjenjuje akcija Reduciraj. S vrha stoga se uzme odgovarajući broj znakova koji pripadaju desnoj strani produkcije, a na vrh stoga se stavi nezavršni znak s lijeve strane produkcije.
- **Akcija Prihvati:**
Ako su pročitani svi znakovi ulaznog niza w , tj. ako je parser pročitao oznaku kraja ulaznog niza \perp , a na stogu je početni nezavršni znak i oznaka dna stoga, parser prihvata niz w .
- **Akcija Odbaci:**
Akcija Odbaci se primjenjuje ako nije moguća daljnja gradnja stabla. Ulazni niz w se ne prihvata odnosno niz w nije u jeziku koji generira zadana gramatika. Parser primjenjuje akciju Odbaci u 2 slučaja:
 - Ako nije zadana nijedna akcija za pročitani znak ulaznog niza i znak vrha stoga
 - Ako znakovi na vrhu stoga nisu jednaki znakovima nijedne desne strane produkcije.

PARSIRANJE TEHNIKOM *POMAKNI-REDUCIRAJ*

Zbog učestale usporedbe znakova na vrhu stoga i znakova desnih strana produkcija, tehnika *Pomakni/Reduciraj* smanjuje broj usporedbi proširivanjem skupa znakova stoga. Za pojedine znakove gramatike definira se više različitih kôdova koji se zapisuju u Tablicu sa kodovima znakova, te se zbog toga na temelju samo jednog znaka na vrhu stoga može odrediti niz znakova na vrhu stoga. Zahvaljujući tome akciju *Reduciraj* moguće je primijeniti bez potrebe čitanja znakova na vrhu stoga i njihove usporedbe sa znakovima desnih strana produkcija.

Kôdovi znakova određuju se za sve znakove gramatike, a označavaju se indeksima. Takvi kodirani znakovi gramatike čine skup znakova stoga. Kodovi znakova jednoznačno određuju dva podatka: znak gramatike i niz znakova gramatike na vrhu stoga.

Parserom *Pomakni/Reduciraj* se upravlja primjenom dviju tablica:

- Tablica *Pomakni/Reduciraj*:
Na temelju kodiranog znaka na vrhu stoga i pročitano znaka ulaznog niza odlučuje se o primjeni akcije *Pomakni* ili *Reduciraj*
- Tablica *Stavi*:
Tablica stavi određuje kod znaka koji se stavlja na vrh stoga.
Retci tablice *Stavi* označeni su znakovima stoga, a stupci znakovima gramatike. Ispunjeni elementi tablice određuju kod znaka koji se stavlja na vrh stoga, dok prazni elementi označavaju akciju *Odbaci*.

Tablica *Stavi* gradi se na sljedeći način. Ako se nadovezivanjem znaka koji se stavlja na vrh stoga s nizom znakova na stogu dobije prefiks jedne od desnih strana produkcije, onda se na vrh stoga stavlja određeni kod znaka.

Tablica *Pomakni/Reduciraj* gradi se na temelju tablice *Stavi*, gramatike te tablice sa kodovima znakova. Akcija *Reduciraj* primjenjuje se samo na one znakove koji su krajnje desni znak desnih strana svih produkcija, a kodove tih znakova moguće je očitati iz tablice sa svim kodovima znakova. Akcija *Prihvati()* stavlja se samo za one kodove znakova koji sadrže znak praznog stoga (∇) i prvi nezavršni znak gramatike u stupcu gdje je ulazni znak \perp . Akcija *Odbaci()* stavlja se za sve znakove stoga gdje je znak ulaznog niza \perp , osim za znak koji zadovoljava uvijete za akciju *Prihvati()*. Akcija *Pomakni()* se stavlja na sva ostala prazna mjesta u tablici.

OSTVARENJE

Zadatak je programski ostvaren u jeziku C# korištenjem alata Microsoft Visual Studio 2008. Ideja zadatka je da program provjerava dali je uneseni niz u jeziku koji generira zadana gramatika.

Program u svom radu preuzme niz koji je korisnik upisao te ga parsira pozivajući funkciju `parsirajUlazniNizZadanomGramatikom`.

```
string ulazniNiz = gramUlazniNizBox.Text;
ulazniNiz += "T"; //Kraj niza
string text = parsirajUlazniNizZadanomGramatikom (gramatika,
tablicaStavi, tablicaPomakniReduciraj, ulazniNiz);
```

Fukcija `parsirajUlazniNizZadanomGramatikom` u svom radu koristi tablicu `Stavi` i tablicu `Pomakni/Reduciraj` koje ovisno o svojim vrijednostima u svakom koraku upravljaju radom

```
int lokacijaUlaznogZnaka = pronadiMjestoZaUlazniZnak
(znakUlaznogNiza.ToString(), tablicaPomakniReduciraj);
int lokacijaZnakaSaStoga = pronadiMjestoZaZnakStoga(znakStoga,
tablicaPomRed);
string akcija = tablicaPomRed [lokacijaZnakaSaStogaPR,
lokacijaUlaznogZnaka];
```

parsera.

Radi lakšeg korištenja programa, tablice su prethodno unešene u program u obliku matrica, a njihovo ostvarenje je opisano u narednim poglavljima.

Akcije parsera zapisane u tablici `PomakniReduciraj` upravljaju načinom rada parsera te o

```
//REDUKCIJA:
if (akcija.Substring(0, 1) == "R")
{
    int reduciraj = int.Parse(akcija.Substring(1, 1));
    for (int j = 0; j < reduciraj; j++)
    {
        string niz = stog.Pop();
        ...
    }
    ...
}
```

```

...
int lokUlZnaka = pronadiMjestoZaUlazniZnak(znakKojemPripadaReduciraniNiz,
tablicaStaviMx)
int lokZnStg = pronadiMjestoZaZnakStoga(noviZnakStoga, tablicaStaviMx);

stog.Push(tablicaStaviMx[lokZnStg, lokUlZnaka]);

```

```

//POMAKNI:
else if (akcija == "P")
{
    int lokacijaUlaznogZnakaSTAVI =
    pronadiMjestoZaUlazniZnak(znakUlaznogNiza.ToString(), tablicaStaviMx);
    int lokacijaZnakaSaStogaSTAVI = pronadiMjestoZaZnakStoga(znakStoga,
    tablicaStaviMx);
    ...
    stog.Push(tablicaStaviMx[lokacijaZnakaSaStogaSTAVI,
    lokacijaUlaznogZnakaSTAVI]);
}

```

njegovim akcijama nad nizom i stogom.

Očitavanja vrijednosti iz tablica ostvareno je dvijema funkcijama: `pronadiMjestoZaUlazniZnak` te `pronadiMjestoZaZnakStoga`. Funkcija `pronadiMjestoZaUlazniZnak` vraća indeks tablice za predani znak ulaznog niza, dok funkcija `pronadiMjestoZaZnakStoga` vraća indeks tablice za predani znak stoga. Na temelju ta dva indeksa moguće je u tablici `Stavi` očitati željeni znak te u tablici `Pomakni/Reduciraj` željenu akciju za trenutne znakove ulaza i stoga. Funkcije su programski ostvarene na sljedeći način:

```

public int pronadiMjestoZaUlazniZnak(string znak, string[,] tablicaStavi)
{
    ...
    for (brojac = 1; brojac < tablicaPretrazivanja.GetLength(1); brojac++)
    {
        if (tablicaPretrazivanja[0, brojac] == znakZaNaci)
        {
            lokacijaUlaznogZnaka = brojac;
            break;
        }
    }
    return lokacijaUlaznogZnaka;
}

```

```

public int pronadiMjestoZaZnakStoga(string znak, string[,] tablicaStavi)
{
    ...
    for (brojac = 1; brojac <= velicina; brojac++)
    {
        if (tablicaPretrazivanja[brojac, 0] == znakZaNaci)
        {
            lokacijaZnakaSaStoga = brojac;
            break;
        }
    }
    ...
    return lokacijaZnakaSaStoga;
}

```

Za korištenje parsera potrebno je izgraditi gramatike te sve sa njima sve potrebne tablice koje parser *Pomakni/Reduciraj* koristi. U sklopu ovog zadatka korištene su tri različite gramatike (G1, G2, G3) za koje je opisan postupak izgradnje tablica.

GRAMATIKA G1:

Produkcije:

- | | | | |
|----|-----|---|---------------|
| 1) | <S> | → | (<A> <S>) |
| 2) | <S> | → | (b) |
| 3) | <A> | → | (<S> a <A>) |
| 4) | <A> | → | (a) |

Za zadanu gramatiku definira se 14 različitih kodova ($a_1, a_2, b_1, (1,)_1,)_2,)_3,)_4, <S>_1, <S>_2, <S>_3, <A>_1, <A>_2, \nabla$), koji se određuju na temelju desnih strana produkcija gramatike.

Tablica sa svim kodovima znakova gramatike:

Znak gramatike	Kod znaka	Niz koji se kodira
a	a ₁	(<S> a
	a ₂	(a
b	b ₁	(b
((₁	(
)) ₁	(<A> <S>)
) ₂	(b)
) ₃	(<S> a <A>)
) ₄	(a)
<S>	<S> ₁	(<A> <S>
	<S> ₂	(<S>
	<S> ₃	∇ <S>
<A>	<A> ₁	(<A>
	<A> ₂	(<S> a <A>
	∇	∇

Nakon popunjavanja tablice sa kodovima znakova imamo sve znakove stoga i može se napraviti tablica *Stavi*.

	a	b	()	<S>	<A>
a ₁			(₁			<A> ₂
a ₂			(₁) ₄		
b ₁			(₁) ₂		
(₁	a ₂	b ₁	(₁		<S> ₂	<A> ₁
<S> ₁			(₁) ₁		
<S> ₂	a ₁		(₁			
<S> ₃			(₁			
<A> ₁			(₁		<S> ₁	
<A> ₂			(₁) ₃		
∇			(₁		<S> ₃	

Budući da se akcija *Reduciraj* primjenjuje samo na znak zatvorene zagrade, jer je to jedini krajnje desni znak desnih strana svih produkcija, tablica *Pomakni/Reduciraj* ima sljedeći sadržaj:

	(a	b)	T
a₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
a₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
b₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
(₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
)₁	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()
)₂	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()
)₃	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()
)₄	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()
<S>₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<S>₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<S>₃	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Prihvati()
<A>₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<A>₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
∇	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()

Akcije ove tablice se mogu definirati ovako:

```
Pomakni(
{
    VrhStoga = Stavi [ZnakNaVrhuStoga, UlazniZnak];
    Pomakni KAZALJKU ulaznog spremnika na sljedeći znak;
}
```

```
Reduciraj1()
{
    UzmiSVrhaStoga (4 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, <S>];
}
```

```
Reduciraj2()
{
    UzmiSVrhaStoga (3 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, <S>];
}
```

```
Reduciraj3()
{
    UzmiSVrhaStoga (5 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, <A>];
}
```

```
Reduciraj4()
{
    UzmiSVrhaStoga (3 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, <A>];
}
```

GRAMATIKA G2

Produkcije:

- 1) $\langle S \rangle \rightarrow (b \langle A \rangle \langle S \rangle \langle B \rangle)$
- 2) $\langle S \rangle \rightarrow (b \langle A \rangle)$
- 3) $\langle A \rangle \rightarrow (d \langle S \rangle c a)$
- 4) $\langle A \rangle \rightarrow (e)$
- 5) $\langle B \rangle \rightarrow (c \langle A \rangle a)$
- 6) $\langle B \rangle \rightarrow (c)$

Za zadanu gramatiku generiraju se 20 različitih kodova: $a_1, a_2, b_1, c_1, c_2, d_1, e_1, (_1,)_1,)_2,)_3,)_4,)_5,)_6, \langle S \rangle_1, \langle S \rangle_2, \langle S \rangle_3, \langle A \rangle_1, \langle A \rangle_2, \langle B \rangle_1, \nabla$

Tablica sa svim kodovima znakova gramatike:

Znak gramatike	Kod znaka	Niz koji se kodira
a	a_1	$(d \langle S \rangle c a$
	a_2	$(c \langle A \rangle a$
b	b_1	$(b$
c	c_1	$(d \langle S \rangle c$
	c_2	$(c$
d	d_1	$(d$
e	e_1	$(e$
($(_1$	$($
)	$)_1$	$(b \langle A \rangle \langle S \rangle \langle B \rangle)$
	$)_2$	$(b \langle A \rangle)$
	$)_3$	$(d \langle S \rangle c a)$
	$)_4$	(e)
	$)_5$	$(c \langle A \rangle a)$
	$)_6$	(c)
$\langle S \rangle$	$\langle S \rangle_1$	$(b \langle A \rangle \langle S \rangle$
	$\langle S \rangle_2$	$(d \langle S \rangle$
	$\langle S \rangle_3$	$\nabla \langle S \rangle$
$\langle A \rangle$	$\langle A \rangle_1$	$(b \langle A \rangle$
	$\langle A \rangle_2$	$(c \langle A \rangle$
$\langle B \rangle$	$\langle B \rangle_1$	$(b \langle A \rangle \langle S \rangle \langle B \rangle$
	∇	∇

Nakon popunjavanja tablice sa kodovima znakova imamo sve znakove stoga i može se napraviti tablica *Stavi*.

	a	b	c	d	e	()	<S>	<A>	
a ₁						(₁) ₃			
a ₂						(₁) ₅			
b ₁						(₁			<A> ₁	
c ₁	a ₁					(₁				
c ₂						(₁) ₆		<A> ₂	
d ₁						(₁		<S> ₂		
e ₁						(₁) ₄			
(₁		b ₁	c ₂	d ₁	e ₁	(₁				
<S> ₁						(₁				 ₁
<S> ₂			c ₁			(₁				
<S> ₃						(₁				
<A> ₁						(₁) ₂	<S> ₁		
<A> ₂	a ₂					(₁				
 ₁						(₁) ₁			
∇						(₁		<S> ₃		

Budući da se akcija *Reduciraj* primjenjuje samo na znak zatvorene zagrade, jer je to jedini krajnje desni znak desnih strana svih produkcija, tablica *Pomakni/Reduciraj* ima sljedeći sadržaj:

	(a	b)	T
a ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
a ₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
b ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
c ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
c ₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
d ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
e ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
(₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
) ₁	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()
) ₂	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()
) ₃	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()
) ₄	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()
) ₅	Reduciraj5()	Reduciraj5()	Reduciraj5()	Reduciraj5()	Reduciraj5()
) ₆	Reduciraj6()	Reduciraj6()	Reduciraj6()	Reduciraj6()	Reduciraj6()
<S> ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<S> ₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<S> ₃	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Prihvati()
<A> ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
<A> ₂	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
 ₁	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()
∇	Pomakni()	Pomakni()	Pomakni()	Pomakni()	Odbaci()

Akcije ove tablice se mogu definirati ovako:

```
Pomakni(  
{  
    VrhStoga = Stavi [ZnakNaVrhuStoga, UlazniZnak];  
    Pomakni KAZALJKU ulaznog spremnika na sljedeći znak;  
}  
  
Reduciraj1()  
{  
    UzmiSVrhaStoga (6 znaka);  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <S>];  
}  
  
Reduciraj2()  
{  
    UzmiSVrhaStoga (4 znaka);  
  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <S>];  
}  
Reduciraj3()  
{  
    UzmiSVrhaStoga (6 znaka);  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <A>];  
}  
  
Reduciraj4()  
{  
    UzmiSVrhaStoga (3 znaka);  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <A>];  
}  
  
Reduciraj5()  
{  
    UzmiSVrhaStoga (5 znaka);  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <B>];  
}  
  
Reduciraj6()  
{  
    UzmiSVrhaStoga (3 znaka);  
    VrhStoga = Stavi [ZnakNaVrhuStoga, <B>];  
}
```

GRAMATIKA G3

Produkcije:

1) $S \rightarrow (L)$

2) $S \rightarrow x$

3) $L \rightarrow S$

4) $L \rightarrow L, S$

Za zadanu gramatiku generiraju se 10 različitih kodova: $x_1, ({}_1,)_1, ,_1, S_1, S_2, S_3, L_1, L_2, \nabla$

Tablica sa svim kodovima znakova gramatike:

Znak gramatike	Kod znaka	Niz koji se kodira
x	x_1	x
($({}_1$	(
)	$)_1$	(L)
,	$,_1$	L ,
S	S_1	S
	S_2	L , S
	S_3	$\nabla <S>$
L	L_1	(L
	L_2	L
	∇	∇

Nakon popunjavanja tablice sa kodovima znakova imamo sve znakove stoga i može se napraviti tablica *Stavi*.

	()	x	S	L	,
$({}_1$	$({}_1$		x_1	S_1	L_1	$<A>_2$
$,_1$	$({}_1$		x_1	S_1		
S_3	$({}_1$		x_1	S_1		
L_1	$({}_1$	$)_1$	x_1	S_1		$,_1$
L_2	$({}_1$		x_1	S_1		$,_1$
∇	$({}_1$		x_1	S_1		

Tablica Pomakni/Reduciraj ima sljedeći oblik:

	()	x	,	T
x_1	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()	Reduciraj1()
(₁	Pomakni()		Pomakni()		
) ₁	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()	Reduciraj2()
, ₁	Pomakni()		Pomakni()		
S_1	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()	Reduciraj3()
S_2	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()	Reduciraj4()
S_3					Prihvati()
L_1	Pomakni()	Pomakni()		Pomakni()	
L_2		Pomakni()		Pomakni()	
∇	Pomakni()		Pomakni()		

Nepopunjeni elementi označavaju akciju Odbaci(). Ostale akcije imaju sljedeći oblik.

Pomakni()

```
{
    VrhStoga = Stavi [ZnakNaVrhuStoga, UlazniZnak];
    Pomakni KAZALJKU ulaznog spremnika na sljedeći znak;
}
```

Reduciraj1()

```
{
    UzmiSVrhaStoga (1 znak);
    VrhStoga = Stavi [ZnakNaVrhuStoga, S];
}
```

Reduciraj2()

```
{
    UzmiSVrhaStoga (3 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, S];
}
```

Reduciraj3()

```
{
    UzmiSVrhaStoga (1 znak);
    VrhStoga = Stavi [ZnakNaVrhuStoga, L];
}
```

Reduciraj4()

```
{
    UzmiSVrhaStoga (3 znaka);
    VrhStoga = Stavi [ZnakNaVrhuStoga, L];
}
```

NAČIN KORIŠTENJA PROGRAMA I PRIMJER IZVOĐENJA

U sklopu programskog ostvarenja napravljeno je i grafičko sučelje za njegovo korištenje.

The screenshot shows a Windows-style application window titled 'Form1'. It contains three vertically stacked sections, each for a different grammar. Each section has a list of grammar rules on the left, an input field labeled 'Ulazni niz:' in the middle, and a 'PARSIRAJ' button on the right.

GRAMATIKA 1:

- $\langle S \rangle \rightarrow (\langle A \rangle \langle S \rangle)$
- $\langle S \rangle \rightarrow (b)$
- $\langle A \rangle \rightarrow (\langle S \rangle a \langle A \rangle)$
- $\langle A \rangle \rightarrow (a)$

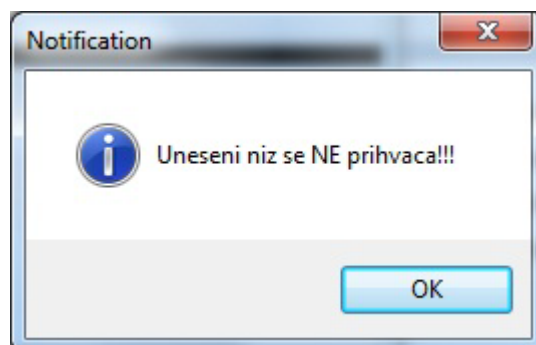
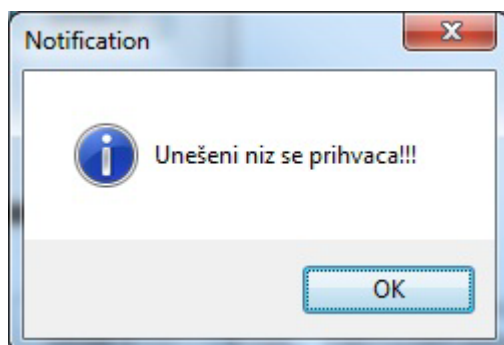
GRAMATIKA 2:

- $\langle S \rangle \rightarrow (b \langle A \rangle \langle S \rangle \langle B \rangle)$
- $\langle S \rangle \rightarrow (b \langle A \rangle)$
- $\langle A \rangle \rightarrow (d \langle S \rangle c a)$
- $\langle A \rangle \rightarrow (e)$
- $\langle B \rangle \rightarrow (c \langle A \rangle a)$
- $\langle B \rangle \rightarrow (c)$

GRAMATIKA 3:

- $S \rightarrow (L)$
- $S \rightarrow x$
- $L \rightarrow S$
- $L \rightarrow L, S$

Sučelje je podijeljeno na tri dijela, svaki za svoju gramatiku. Pritiskom na jednom od tri gumba, niz koji je upisan u kućici „Ulazni niz“ se parsira prema pravilima gramatike sa lijeve strane. Ako je uneseni niz dobro zadan, tj. ako je niz u jeziku koji prihvaća gramatika program će javiti poruku da se nize prihvaća, a ako je niz pogrešno zadan program će javiti grešku te da se zadani niz ne prihvaća.



Primjer pravilno napravljenih ulaznih nizova:

Za gramatiku G1:

- 1) $\langle S \rangle \rightarrow (\langle A \rangle \langle S \rangle)$
- 2) $\langle S \rangle \rightarrow ((\langle S \rangle a \langle A \rangle) (b))$
- 3) $\langle S \rangle \rightarrow (((b) a (a)) (b))$

Korištene produkcije:

- 1)
- 2) $\langle A \rangle \rightarrow (\langle S \rangle a \langle A \rangle)$, $\langle S \rangle \rightarrow (b)$
- 3) $\langle S \rangle \rightarrow (a)$

Za gramatiku G2:

- 1) $\langle S \rangle \rightarrow (b \langle A \rangle \langle S \rangle \langle B \rangle)$
- 2) $\langle S \rangle \rightarrow (b (e) \langle S \rangle \langle B \rangle)$
- 3) $\langle S \rangle \rightarrow (b (e) (b \langle A \rangle) \langle B \rangle)$
- 4) $\langle S \rangle \rightarrow (b (e) (b (e)) (c))$

Korištene produkcije:

- 1)
- 2) $\langle A \rangle \rightarrow (e)$
- 3) $\langle S \rangle \rightarrow (b \langle A \rangle)$
- 4) $\langle A \rangle \rightarrow (e)$, $\langle B \rangle \rightarrow (c)$

Za gramatiku G3:

- 1) $S \rightarrow (L)$
- 2) $S \rightarrow (L , S)$
- 3) $S \rightarrow (S , (L))$
- 4) $S \rightarrow (S , (S))$
- 5) $S \rightarrow (x , (x))$

Korištene produkcije:

- 1)
- 2) $L \rightarrow L , S$
- 3) $L \rightarrow S$, $S \rightarrow (L)$
- 4) $L \rightarrow S$
- 5) $S \rightarrow x$

Primjer korištenja programa:

Form1

GRAMATIKA 1:

$\langle S \rangle \rightarrow (\langle A \rangle \langle S \rangle)$
 $\langle S \rangle \rightarrow (b)$
 $\langle A \rangle \rightarrow (\langle S \rangle a \langle A \rangle)$
 $\langle A \rangle \rightarrow (a)$

Ulazni niz:

Unešeni niz se prihvaca

PARSIRAJ GRAM 1

GRAMATIKA 2:

$\langle S \rangle \rightarrow (b \langle A \rangle \langle S \rangle \langle B \rangle)$
 $\langle S \rangle \rightarrow (b \langle A \rangle)$
 $\langle A \rangle \rightarrow (d \langle S \rangle c a)$
 $\langle A \rangle \rightarrow (e)$
 $\langle B \rangle \rightarrow (c \langle A \rangle a)$
 $\langle B \rangle \rightarrow (c)$

Ulazni niz:

Unešeni niz se prihvaca

PARSIRAJ GRAM 2

GRAMATIKA 3:

$S \rightarrow (L)$
 $S \rightarrow x$
 $L \rightarrow S$
 $L \rightarrow L, S$

Ulazni niz:

Unešeni niz se prihvaca

PARSIRAJ GRAM 3

Form1

GRAMATIKA 1:

$\langle S \rangle \rightarrow (\langle A \rangle \langle S \rangle)$
 $\langle S \rangle \rightarrow (b)$
 $\langle A \rangle \rightarrow (\langle S \rangle a \langle A \rangle)$
 $\langle A \rangle \rightarrow (a)$

Ulazni niz:

Unešeni niz se NE prihvaca

PARSIRAJ GRAM 1

GRAMATIKA 2:

$\langle S \rangle \rightarrow (b \langle A \rangle \langle S \rangle \langle B \rangle)$
 $\langle S \rangle \rightarrow (b \langle A \rangle)$
 $\langle A \rangle \rightarrow (d \langle S \rangle c a)$
 $\langle A \rangle \rightarrow (e)$
 $\langle B \rangle \rightarrow (c \langle A \rangle a)$
 $\langle B \rangle \rightarrow (c)$

Ulazni niz:

Unešeni niz se NE prihvaca

PARSIRAJ GRAM 2

GRAMATIKA 3:

$S \rightarrow (L)$
 $S \rightarrow x$
 $L \rightarrow S$
 $L \rightarrow L, S$

Ulazni niz:

Unešeni niz se NE prihvaca

PARSIRAJ GRAM 3

ZAKLJUČAK

U ovom samostalnom studentskom projektu bilo je potrebno ostvariti funkcionalno programsko rješenje parsera *Pomakni/Reduciraj*. To je učinjeno korištenjem alata Microsoft Visual Studio 2008, te je rješenje napisano u programskom jeziku C#. Rezultat je program SSP koji uspješno provjerava ispravnost zadanog niza.

Opisani postupak parsiranja je učinkoviti, jer su sve odluke o primjeni akcija jednoznačne. Osim toga, jednoznačna je i odluka koju je od redukcija potrebno primjeniti. Međutim, razliku između tehnika parsiranja moguće je najbolje uočiti u slučaju neprihvatanja niza. Jednostavniji parser utvrđuje grešku tijekom ispitivanja da li je vrh stoga desna strana produkcija, dok parser *Pomakni/Reduciraj* utvrđuje pogrešku u trenutku stavljanja znakova na stog koji nisu prefiks desne strane nijedne produkcije zadane gramatike.

Korištenje samog parsera se pokušalo pojednostaviti grafičkim sučeljem koji jednostavnim dizajnom i intuitivnim korištenjem uveliko olakšava posao korisnika.