

PROGRAMSKI PREVODIOCI

Drugi projekat

Projektni zadatak se sastoji iz četiri dela:

1. Gramatika 5p
2. Editor 5p
3. Interpreter 10p
4. Debager 10p

Gramatika

Na sledećem linku predstavljeni su jednostavni programerski zadaci rešeni na izmišljenom programskom jeziku:

https://docs.google.com/document/d/12LQYbAITPXJIXbIGC8wfgaRaRW_BrybJwT1Ge9_WqpM/edit?usp=sharing

Zadatak prvog dela je napisati gramatiku koja opisuje pomenuti jezik. U pitanju je CFG koju treba predstaviti formalno, sa akcentom na skupu produkcija. Gramatiku napisati u formi po želji. To može da bude npr. Word fajl ili fotografisan papir.

Editor

Implementirati editor za pisanje programa na pomenutom jeziku iz prvog dela. Za ovaj deo zadatka, najvažnija funkcionalnost editora je 'syntax highlighting', odnosno mogućnost editora da pojedine ključne reči iz jezika oboji u neku boju onog trenutka kada ta reč bude otkucana. Izbor skupa ključnih reči i njihovih boja nije određen. Navažnije je pokazati funkcionalnost na par primera.

Interpreter

Potrebno je omogućiti izvršavanje programa napisanom na našem izmišljenom jeziku.

Ako program nije pravilno napisan, potrebno je o tome napisati odgovarajuću poruku. Neophodno je proveravati samo situacije u kojima može doći do greške u vremenu kompilacije. Greške o kojima treba voditi računa su:

- Sve sintaksne greške: nailaženje na neočekivni tip tokena
- Definisanje iste varijable više od jedanput
- Pozvana varijabla nije definisana u opsegu vidljivosti, trenutnom ili nekom od obuhvatajućih
- Poziv funkcije koja nije definisana ili poziv ugrađene funkcije koja ne postoji
- Definisanje iste funkcije više od jedanput, slično kao sa varijablama
- Pogrešan broj parametara pri pozivu funkcije
- Pogrešan tip parametara pri pozivu funkcije
- Pogrešna povratna vrednost funkcije
- Uparivanje različitih tipova podataka bez mogućnosti konverzije

Napomena na neke specifičnosti jezika:

- U jeziku postoje ugrađene i 'custom' definisane, tj. korisničke funkcije. Ugrađene funkcije treba da budu 'hardkodovane' u interpreter, tj. kada se na primer pri interpretaciji naiđe na funkciju za učitavanje sa konzole, jednostavno treba upotrebiti određenu funkciju jezika u kome pišete interpreter. Sa druge strane pri izvršavanju korisničkih funkcija treba potražiti njihovu implementaciju.

- U jeziku postoji tip podataka koji predstavlja kolekcije. To su ujedno i liste i stringovi. Ovaj tip podataka u našem jeziku ima mogućnost dohvaćanja elemenata po indeksu, kao i mogućnost pravljenja višedimenzionalnih kolekcija. Veličina kolekcije se ne navodi pri deklaraciji, a dodavanje novog elementa se postiže dodavanjem na postojeći indeks ili prvi sledeći indeks posle poslednjeg. Na primer, ako kolekcija trenutno ima 5 elemenata (indeksiranih od 0 do 4), novi element može da se ubaci u kolekciju dodavanjem na indeks 5. Testiraće se samo ispravni slučajevi ubacivanja i dohvaćanja elemenata iz kolekcije.
- Od funkcionalnosti interpretera treba pokriti samo one koje su navedene u primerima, to su različiti tipovi podataka, rad sa funkcijama, uslovi, petlje, rad sa varijablama i osnovni matematički izrazi.
- Ako se u nekoj operaciji jave operandi različitog tipa, jedan je potrebno konvertovati, na primer u funkciji koja konkatira string, ako se prosledi broj, njega je potrebno pretvoriti u string.

Projekat možete testirati na primerima pomenutim u prvom delu. U slučaju da u primerima postoji greška, možete ostaviti komentar sa objašnjenjem, čime biste osvojili nagradni poen.

Postoji alternativna mogućnost realizacije ovog dela zadatka, u kojoj je potrebno generisati sliku apstraktnog sintaksnog stabla, slično kao što je rađeno na vežbama. Ova mogućnost bi donosila najviše 5 poena.

Debager

Zadatak je implementirati debager za pomenuti izmišljeni jezik. Potrebno je omogućiti da se nad nekim instrukcijama obeleži breakpoint gde bi se pri pokretanju programa u debug modu izvršavanje zaustavilo. Takođe je potrebno pri zaustavljanju na određenoj instrukciji prikazati stanje vidljivih varijabli: u tom trenutku koje varijable su deklarisanе i vidljive iz trenutnog opsega i koja je njihova trenutna vrednost.

Nakon zaustavljanja programa na određenom breakpoint-u treba omogućiti tri tipa nastavljanja izvršavanja:

- Sledeća instrukcija bez ulaska u blok te instrukcije
- Sledeća instrukcija sa ulaskom u blok instrukcije
- Sledeći breakpoint

Pokretanje debug moda se može izvesti preko editora, ukoliko postoji, pri čemu bi se sa leve strane mogle obeležiti instrukcije nad kojima se postavlja breakpoint. Breakpoint se može postaviti samo na početak instrukcije.

Rok i odbrana

Rok za izradu projektnog zadatka je termin u kolokvijumskoj nedelji.

Za izradu zadatka je dozvoljeno koristiti tehnologiju i jezik po izboru, kao i upotreba primera sa interneta i sa vežbi. Ukoliko koristite primer sa interneta kao polaznu tačku, na odbrani morate objasniti kako taj primer funkcioniše, inače neće biti priznat.