

Snail dokumentacija

Barbara Posavec, Ivona Čižić, Luka Jandrijević

June 2022

1 Lekser

Napravila: Barbara Posavec

1.1 Tipovi tokena

Lekser se sastoji od različitih simbola za usporedbu ($<$, $>$, $>=$, $<=$, $!=$...) i pridruživanje je označeno s znakom jednakosti $=$. U snailu su ključne riječi "if", "else", "then", "print", "endif", "newline" pa za njih postoje posebni tokeni. U zadaći su dodani i još dodatni tipovi poput "return", "call", "def" i "inpt" koji će služiti za vraćanje vrijednosti u funkciji, pozivanje i definiranje funkcija te "inpt" će služiti za unošenje brojeva u varijablu s tipkovnice. Također smo definirali i tokene za imena varijable (IME), za tekst te za broj. Tip podatka koji nije uobičajen u poznatim programskim jezicima (C, C++, Javascript itd...) je razlomak koji smo ovdje definirali kao token tipa BROJ.

Drugi tip podatka je konačan. Njega označujemo sa da, ne ili možda te je sličan kao bool u pythonu. Vrijednost 'da' je 1, vrijednost 'ne' je 0 a vrijednost tipa 'možda' je -1.

Za unarne operatore odabrali smo $++$ i $-$ koji povećavaju/umanjuju tip podatka za jedan.

Za ternarnu operaciju smo odabrali znak '?' koji čita sljedeća tri broja te vraća 0 ako je bilo koji broj od ta 3 broja jednak nuli a inače vraća 1

1.2 snail lexer

Lexer smo nazvali snail. U njemu čitamo znakove i ako naiđemo na prazninu tada ju ignoriramo. Ako pročitamo znak 'i' tada ovisno o tome koji je znak sljedeći vraćamo T.MANJE ili T.MANJEJ. Slično radimo za ostale znakove. Znamo da smo naišli na neki tekst ako pročitamo navodnike te onda čitamo sve znakove dok opet ne naiđemo na navodnike koje označavaju kraj teksta/stringa. Ako pročitamo neki znak onda čitamo znakove ili brojeve te znamo da je to tada token za ime varijable/funkcije. Ako naiđemo na znak '/' i nakon toga opet na znak '/' onda znamo da je to jednolinijski komentar i zanemarujemo sav tekst u tome redu. Višelinijski komentari su označeni znakom '#' na početku i kraju.

2 Parser

napravio: Luka Jandrijević

Beskontekstna gramatika je napisana u kodu. Početna varijabla je `naredbe_lista` pomoću koje generiramo proizvoljno varijabli imena "naredbe". "naredbe" zatim mogu biti ili pridruživanje, `print`, `if` ili unos varijable. Naš program se sastoji od niza funkcija i u svakoj od njih je niz naredbi. Program počinje od funkcije `main()` te ona mora biti definirana u svakom programu.

U funkciji program čitamo i spremamo sve definicije funkcija te pozivamo funkciju `p.funkcija()` koja se dalje brine o naredbama. Funkcija `naredbe_lista` služi da u funkciji ili `if` naredbi bude dozvoljeno više naredbi. U funkciji `naredbe()`, ovisno o tokenu pozivamo određenu funkciju ili AST koji će se dalje brinuti o tome što se mora obaviti.

Bitna funkcija je možda `poziv()` koja služi da poziva funkciju ako se radi o funkciji ili da vrati vrijednost varijable ako se radi o varijabli. Naime i varijabla i funkcija počinju s tokenom `T.IME` pa kada pročitamo takav token i dalje neznamo je li se radi o funkciji ili o varijabli.

Funkcije `print_naredba()`, `if_naredba()`, `intp()` brinu o tome da pozivaju određenu AST klasu s danim parametrima. Funkcija `parametri()` vraćaju listu parametara koji se zadaju funkciji. Oni se čitaju od otvorene zagrade do zatvorene zagrade.

U funkciji `broj()` prvo pozivamo `p.aritm()` jer u BKG znamo da na prvom mjestu dolazi aritm. Zatim gledamo jesmo li naišli na neku od usporedba te ako jesmo pozivamo određeni AST a ako naiđemo na unarni token onda pozivamo određeni AST. Inače iza aritm ne dolazi ništa pa znamo da je on samostalan i vraćamo ga.

Funkcije `aritm()`, `član()`, `faktor()` su slične kao i na predavanjima.

3 AST

napravila: Ivona Čižić

U klasi suprotan, funkcija vrijednost vraća vrijednost starog broja sa minusom ispred. Ternarni je operator koji vraća 0 ako je bilo koji od 3 člana jednaka 0, inače vraća 1. U `Vrati` koristimo memoriju kako bi vratili vrijednost varijable/funkcije. Funkcija služi da definiramo i izvršimo tijelo funkcije koju smo prethodno definirali.

U `Pozivu` se brinemo za pozivanje funkcije i za rekurzivne pozive. `Inpt` opet koristi Memoriju kako bi zapamtila vrijednost koju je korisnik upisao u konzoli. `Pridruživanje` je trivijalno.

If gleda vrijednost uvjeta te ovisno o uvjetu izvršava naredbe u grani. `Print` prvo gleda koja vrsta tokena se ispisuje i sukladno tome ispisuje vrijednost na ekran.

U Usporedbi gledamo lijevu i desnu vrijednost te gledamo koji od tipova usporedbi nije jednak nenavedenom. Sukladno tome vraćamo vrijednost.

Osnovna služi za osnovne operacije množenje, djeljenje itd... Gleda jednostavno koji je token idući te sukladno tome zbraja, oduzima itd...

Unarni je slično kao Osnovno samo što tu imamo 2 tipa za provjeriti. Ili je ++ ili --. Ovisno o tome smanjujemo/povećavamo vrijednost.