

Poročilo

Maj 2025

Kazalo

1	Uvod	3
2	Osnova	3
2.1	Konstrukti	3
2.1.1	Enota	3
2.1.2	Realna števila	3
2.1.3	Nizi	3
2.1.4	Koordinate	4
2.1.5	Bloki	4
2.1.5.1	City	4
2.1.5.2	Road	4
2.1.5.3	Building	4
2.1.5.4	Area	4
2.1.6	Ukazi	4
2.1.6.1	Polyline	4
2.1.6.2	Polygon	4
2.1.6.3	Circle	5
3	Nadgradja	5
3.1	Validacija	5
3.2	Dodatni elementi	5
3.2.1	Lake	5
3.2.2	Park	5
3.3	Spremenljivke	5
3.4	Izjave	5
3.4.1	Seštevanje	5
3.4.2	Odštevanje	5
3.4.3	First & Second	6
3.5	Povpraševanja	6
4	Gramatika z BNF notacijo	6
5	Izračun FIRST in FOLLOW množic	7
5.1	Izračun FIRST množic !	7
5.2	Izračun FOLLOW množic	8
6	Priprava smiselnih testnih primerov	9
6.1	Primer	9
6.2	Primer	9
6.3	Primer	10
6.4	Primer	10
6.5	Primer	11

1 Uvod

Za predmet prevajanje programskih jezikov smo se odločili, da bomo implementirali prevajalnik za naš lastni programski jezik. Ta jezik bo namenjen opisovanju mestne infratrukture in njihovih lastnosti. Naš cilj je ustvariti preprost prevajalnik, ki bo sposoben obdelati osnovne geometrijske konstrukte in jih pretvoriti v GeoJSON format.

2 Osnova

Zahtevano je, da lahko z uporabo vašega jezika opišete geometrijske strukture, točke in ceste v mestu. Torej je potrebno podpreti polilinije (polyline) in poligone (polygon). Na primer, majhen del mesta bi lahko opisali takole:

```
let @p = (1,1.12);
let @q = (2,2);

city ["Maribor"]{
  road["Ptujška cesta"]{
    polyline[(1,2.5),(3.1,4),(5,6)];
  };
  building["FERI"]{
    polygon[(1,1),(1,2),(2,2.7),(2,1),(1,1)];
  };
  area["Igrišče"]{
    polygon[(2,3),(1,1),(5,7),(2,3)];
  };
  lake["Jezero Bled"]{
    circle[(4,3),2.8];
  };
  park["Park Maribor"]{
    circle[(5,9),2];
  };
  road["Ljubljanska cesta"]{
    polyline[$p,$q,$p+$q];
  };
}

?{[(1,1),$p,$q,(3,4)],[(1,1),3]};
let @r = (fst(p),snd(q));
```

2.1 Konstrukti

Uporabljen jezik vsebuje naslednje konstrukte.

2.1.1 Enota

Naša nevtralna enota je: null

2.1.2 Realna števila

Za vrednosti imamo realna števila: 1.0, 2.5, -3.14, 0.0

2.1.3 Nizi

Za nize uporabljamo dvojne narekovaje, najdemo jih lahko v oglatih oklepajih: "Ptujška cesta", "FERI", "Park"

2.1.4 Koordinate

S koordinatami lahko predstavimo lokacije na zemljevidu, v tem primeru je prva komponenta longituda, druga komponenta pa je latituda. (1.0,2.5), (3.1,4), (5,6)

2.1.5 Bloki

Blok je sestavljen iz imena objekta, ki ga želimo definirati, definiramo jih lahko s "polygon", "polyline" in "circle". Blok se zaključi z okroglimi oklepaji.

2.1.5.1 City

Blok "City" je osrednji blok, ki ga uporabljamo za definiranje mesta. Vsebuje lahko druge bloke, kot so ceste, stavbe, območja, parki in jezera.

```
city["IME"]{  
    BLOKI  
};
```

2.1.5.2 Road

"Road" je blok, ki ga uporabljamo za definiranje ceste. Vsebuje lahko ukaze za izris, ki izrišejo črte.

```
road["IME"]{  
    COMMANDS  
};
```

2.1.5.3 Building

"Building" je blok, ki ga uporabljamo za definiranje stavb. Vsebuje lahko ukaze za izris stavbe.

```
building["IME"]{  
    COMMANDS  
};
```

2.1.5.4 Area

"Area" je blok, ki ga uporabljamo za definiranje območij. Vsebuje lahko ukaze za izris območij.

```
area["IME"]{  
    COMMANDS  
};
```

2.1.6 Ukazi

Bloki vsebujejo ukaze, ki jih lahko uporabimo za izris geometrijskih oblik. Ukazi so lahko "polyline", "polygon" in "circle".

2.1.6.1 Polyline

"Polyline" je ukaz, ki ga uporabljamo za izris poliliniij. Vsebuje lahko koordinate za točke, med katerimi so izrisane črte. Končna in začetna točki sta lahko poljubni.

```
polyline[TOČKA,TOČKA,TOČKA];
```

2.1.6.2 Polygon

"Polygon" je ukaz, ki ga uporabljamo za izris poligonov. Vsebuje lahko koordinate za točke, med katerimi so izrisane črte. Poligon se mora zaključiti z začetno točko.

```
polygon[TOČKA1,TOČKA2,TOČKA3,TOČKA1];
```

2.1.6.3 Circle

“Circle” je ukaz, ki ga uporabljamo za izris krogov. Vsebuje lahko koordinato za središče in polmer kroga.

```
circle[TOČKA, RADIJ];
```

3 Nadgradja

Jezik smo nadgradili z validacijo, dodatnimi elementi, spremenljivkami, izjavami in povpraševanji.

3.1 Validacija

Validirali bomo poligone pri čemer bomo preverili, da se prvi in zadnji točki ujemata, pri tem morajo biti vse točke v poligonu različne (razen prve in zadnje). Hkrati bomo preverjali, da se črte ob izrisu poligona ne prekrivajo ali sekajo. Ob pojavitvi napake bomo o tem obvestili uporabnika.

```
polygon[(1,1),(3,3),(3,1),(1,3),(1,1)];
```

3.2 Dodatni elementi

Na zemljevidu lahko definiramo tudi dodatne elemente, ki jih lahko uporabljamo za definicijo mesta.

3.2.1 Lake

“Lake” je blok, ki ga uporabljamo za definiranje jezer. Vsebuje lahko ukaze za izris jezer.

```
lake["Jezero Bled"]{  
    COMMANDS  
};
```

3.2.2 Park

“Park” je blok, ki ga uporabljamo za definiranje parkov. Vsebuje lahko ukaze za izris parkov.

```
park["IME"]{  
    COMMANDS  
};
```

3.3 Spremenljivke

V jeziku lahko definiramo spremenljivke, ki jih lahko uporabljamo za shranjevanje vrednosti. Spremenljivke se definirajo z uporabo ključa “let”, in predstavlja točko, ki jo definiramo v oklepajih. Pred samim imenom spremenljivke mora biti znak \$.

```
let @IME = (KOORDINATA X,KOORDINATA Y);
```

3.4 Izjave

Podprte izjave v jeziku vključujejo: seštevanje, odštevanje, dostop do prve koordinate in dostop do druge koordinate.

3.4.1 Seštevanje

Seštevanje dveh točk se izvede tako, da se seštejeta obe komponenti točk. Rezultat je nova točka.

```
polyline[(3,4),(2,1)+(3,4)];
```

3.4.2 Odštevanje

Odštevanje dveh točk se izvede tako, da se odštejeta obe komponenti točk. Rezultat je nova točka.

```
polyline[(1,1),$q,(2,4)-$q];
```

3.4.3 First & Second

Dostop do prve in druge komponente točke se izvede tako, da se uporabita funkciji `fst` in `snd`.

```
let @r = (fst($p),snd($q));
```

3.5 Povpraševanja

Povpraševanje je določeno z množico točk, zapisanih v oglatih oklepajih. Poleg te množice je podana dodatna točka z določenim radijem, ki opredeljuje krožno območje povpraševanja. Rezultat povpraševanja je množica vseh točk iz začetne množice, ki ležijo znotraj tega krožnega območja.

```
?{[TOČKA1,TOČKA2,TOČKA3,TOČKA4],[TOČKA0,RADIJ]};
```

4 Gramatika z BNF notacijo

```
izraz    ::= izraz** izraz*  
izraz*   ::= izraz izraz* | ε  
izraz**  ::= SPREMENLJIVKA_DEF | POVPRASEVANJE | CITY
```

```
// definicija mesta  
CITY ::= city["IME"] { BLOCKS }
```

```
// definicija bloka  
BLOCKS ::= BLOCK BLOCKS*  
BLOCKS* ::= BLOCK BLOCKS* | ε  
BLOCK  ::= ROAD | BUILDING | AREA | LAKE | PARK
```

```
ROAD ::= road["IME"] { POLYLINE }  
BUILDING ::= building["IME"] { IZRIS }  
AREA  ::= area["IME"] { IZRIS }  
LAKE  ::= lake["IME"] { IZRIS }  
PARK  ::= park["IME"] { IZRIS }
```

```
IZRIS ::= POLYGON | KROG
```

```
// definicija spremenljivke  
SPREMENLJIVKA_DEF ::= let @ IME = TOČKA;
```

```
POLYLINE ::= polyline[TOČKE];  
POLYGON  ::= polygon[TOČKE];
```

```
KROG ::= circle KROG*  
KROG* ::= [TOČKA, KROG**]  
KROG** ::= FIRST_SECOND | ŠTEVILO
```

```
// definicija povpraševanja  
POVPRAŠEVANJE ::= ?{[TOČKE],KROG*};
```

```
//definicija večih točk  
TOČKE ::= TOČKA TOČKE*  
TOČKE* ::= , TOČKA TOČKE* | ε
```

```

// definicija točke
TOČKA ::= TOČKA** TOČKA*
TOČKA* ::= OPERACIJA TOČKA** TOČKA* | ε
TOČKA** ::= ( KOORDINATA , KOORDINATA ) | SPREMENLJIVKA PERACIJA

KOORDINATA ::= ŠTEVILO | FIRST_SECOND

OPERACIJA ::= + | -

// definicija FIRST in SECOND
FIRST_SECOND ::= fst TOČKA | snd TOČKA

// uporaba spremenljivke
SPREMENLJIVKA ::= $IME

ŠTEVILO ::= [0-9] ŠTEVILO*
ŠTEVILO* ::= [0-9] ŠTEVILO* | . REALNO | ε

REALNO ::= [0-9] REALNO*
REALNO* ::= [0-9] REALNO* | ε

// definicija niza
IME ::= [a-zA-Z_] IME*
IME* ::= [a-zA-Z0-9_] IME* | ε

```

5 Izračun FIRST in FOLLOW množic

5.1 Izračun FIRST množic !

```

FIRST(IZRAZ) = { let, ?, city }
FIRST(IZRAZ*) = { let, ?, city, ε }
FIRST(IZRAZ**) = { let, ?, city }
FIRST(CITY) = { city }
FIRST(BLOCKS) = { road, building, area, lake, park }
FIRST(BLOCKS*) = { road, building, area, lake, park, ε }
FIRST(ROAD) = { road }
FIRST(BUILDING) = { building }
FIRST(AREA) = { area }
FIRST(LAKE) = { lake }
FIRST(PARK) = { park }
FIRST(IZRIS) = { polygon, circle }
FIRST(SPREMENLJIVKA_DEF) = { let }
FIRST(POLYLINE) = { polyline }
FIRST(POLYGON) = { polygon }
FIRST(KROG) = { circle }
FIRST(KROG*) = { [ }
FIRST(KROG**) = { fst, snd, [0-9] }
FIRST(POVPRAŠEVANJE) = { ? }
FIRST(TOČKE) = { ( , $ }
FIRST(TOČKE*) = { , ε }
FIRST(TOČKA) = { ( , $ }
FIRST(TOČKA*) = { + , - }

```

FIRST(TOČKA**) = { (, \$ }
 FIRST(KOORDINATA) = { [0-9] , fst , snd}
 FIRST(OPERACIJA) = { + , - }
 FIRST(FIRST_SECOND) = { fst , snd }
 FIRST(SPREMELNJIVKA) = { \$ }
 FIRST (ŠTEVILO) = { [0-9]}
 FIRST (ŠTEVILO*) = { [0-9], . , ε }
 FIRST(REALNO) = { [0-9] }
 FIRST(REALNO*) = { [0-9], ε }
 FIRST(IME) = { [a-zA-Z0-9_] }
 FIRST(IME*) = { [a-zA-Z0-9_], ε }

5.2 Izračun FOLLOW množic

FOLLOW(IZRAZ) = { let, ?, city, ε }
 FOLLOW(IZRAZ*) = { let, ?, city, ε }
 FOLLOW(IZRAZ**) = { let, ?, city, ε }
 FOLLOW(CITY) = { let, ?, city, ε }
 FOLLOW(BLOCKS) = { } }
 FOLLOW(BLOCKS*) = { } }
 FOLLOW(BLOCK) = { road, building, area, lake, park, }
 FOLLOW(ROAD) = { road, building, area, lake, park, }
 FOLLOW(BUILDING) = { road, building, area, lake, park, }
 FOLLOW(AREA) = { road, building, area, lake, park, }
 FOLLOW(LAKE) = { road, building, area, lake, park, }
 FOLLOW(PARK) = { road, building, area, lake, park, }
 FOLLOW(IZRIS) = { } }
 FOLLOW(SPREMELNJIVKA_DEF) = { let, ?, city, ε }
 FOLLOW(POLYLINE) = { } }
 FOLLOW(POLYGON) = { } }
 FOLLOW(KROG) = { } }
 FOLLOW(KROG*) = { } }
 FOLLOW(KROG**) = {] }
 FOLLOW(POVPRAŠEVANJE) = { let, ?, city, ε }
 FOLLOW(TOČKE) = {] }
 FOLLOW(TOČKE*) = {] }
 FOLLOW(TOČKA) = { ; , , ε,)}
 FOLLOW(TOČKA*) = { ; , , ε,)}
 FOLLOW(TOČKA**) = { + , - }
 FOLLOW(KOORDINATA) = { ,) }
 FOLLOW(OPERACIJA) = { TOČKA }
 FOLLOW(FIRST_SECOND) = { ,) }
 FOLLOW(SPREMELNJIVKA) = { + , - , ε }
 FOLLOW(ŠTEVILO) = { ,) }
 FOLLOW(ŠTEVILO*) = { ,) }
 FOLLOW(REALNO) = { ,)}
 FOLLOW(REALNO*) = { ,) }
 FOLLOW(IME) = { let, ?, city, ε }
 FOLLOW(IME*) = { let, ?, city, ε }

6 Priprava smiselnih testnih primerov

6.1 Primer

```
city{
  road["Ptujška cesta"]{
    polyline[(1,2),(3,4),(5,6)];
  };
  area["FERI"]{
    polygon[(1,1),(1,2),(2,2),(2,1),(1,1)];
  };
  area["Park"]{
    circle[(4,3),2];
  };
  let $p = (1,1);
  let $q = (2,2);

  road["Ljubljanska cesta"]{
    polyline[$p,$q,$p+$q];
  };

  ?{[(1,1),$p,$q,(3,4)],[(1,1),3]};

  let $r = (fst(p),snd(q));
```

6.2 Primer

```
let $a = (0,0);
let $b = (1,1);
let $c = (2,2);
let $d = (3,3);

city["Veliko mesto"] {
  road["Glavna ulica"] {
    polyline[$a,$b,$c,$d];
  };

  building["Občina"] {
    polygon[(1,1),(2,1),(2,2),(1,2),(1,1)];
  };

  area["Trg"] {
    polygon[(3,3),(4,3),(4,4),(3,4),(3,3)];
  };

  park["Zeleni park"] {
    circle[(5,5),2.5];
  };

  lake["Veliko jezero"] {
    circle[(6,6),4];
  };
}
```

6.3 Primer

```
let $x = (3,3);
let $y = (6,6);

city["Kompleksno mesto"] {
  road["Severna"] {
    polyline[(0,0), $x, (6,0)];
  };
  building["Muzej"] {
    polygon[(2,2), (4,2), (4,4), (2,4), (2,2)];
  };
  area["Stadion"] {
    polygon[(5,5), (6,6), (7,5), (6,4), (5,5)];
  };
  park["Jugozahodni park"] {
    circle[(1,1), 2];
  };
  lake["Ribnik"] {
    circle[$y, 1.5];
  };
  road["Vzhodna"] {
    polyline[$x, $y, $x+$y];
  };

  ?{[(3,3), $x, $y, (5,5)], [(4,4), 2]};
}
```

6.4 Primer

```
let $a = (1,1);
let $b = (2,3);
let $c = $a + $b;
let $d = (fst($b), snd($c));

city["Ljubljana"]{
  road["Dunajska"]{
    polyline[(1,2), (2,3), $c];
  };
  road["Celovška"]{
    polyline[$a, (2,2), $d];
  };
  building["Modra stavba"]{
    polygon[(1,1), (1,2), (2,2), (2,1), (1,1)];
  };
  area["Trg Republike"]{
    polygon[(1,1), (2,4), (4,4), (4,2), (1,1)];
  };
  lake["Zbiljsko jezero"]{
    circle[(5,5), 2.5];
  };
  park["Park Tivoli"]{
    circle[(4,3), 3];
  };
}
```

```
}
```

```
?{[$a, $b, $c, $d],[ (2,2),2]};  
let $rez = ($a + $b) - (1,1);
```

6.5 Primer

```
let $p1 = (3,3);  
let $p2 = (1,2);  
let $p3 = (2,2);  
let $mid = ($p1 + $p2) - (1,1);  
let $origin = (0,0);  
let $z = (fst($mid), snd($p3));
```

```
city["Koper"]{  
  road["Obalna"]{  
    polyline[$p1, $p2, $p3, $p1];  
  };  
  area["Trg Koper"]{  
    polygon[(1,1), (2,2), (2,4), (1,1)];  
  };  
  building["Zgradba A"]{  
    polygon[(0,0), (0,2), (2,2), (2,0), (0,0)];  
  };  
  lake["Jezero Koper"]{  
    circle[$p2, 1.8];  
  };  
}
```

```
city["Celje"]{  
  road["Kidričeva"]{  
    polyline[(0,0), $mid, $mid + (1,1)];  
  };  
  park["Mestni park"]{  
    circle[$origin, 3];  
  };  
  area["Stari trg"]{  
    polygon[(1,1),(2,2),(3,3),(4,4),(1,1)]; // validacija: pravilno zaprt  
  };  
  building["Dvorana"]{  
    polygon[(1,1),(1,3),(2,3),(2,1),(1,1)];  
  };  
}
```

```
?{[ $p1, $p2, $p3, (4,4) ], [ $p2, 2.0 ]};  
let $v = (fst($p1) + 1, snd($p2) - 1);
```