Dokumentacia

# Navrh (casti prekladaca, predavanie informacii medzi nimi)

# Implementacia(pouzite datove struktury, tabulka symbolov, generovanie kodu)

# Vyvojovy cyklus

# Sposob prace v time

# Specialne pouzite techniky a algoritmy

# Odchylky od prednasanej latky alebo tradicnych pristupov

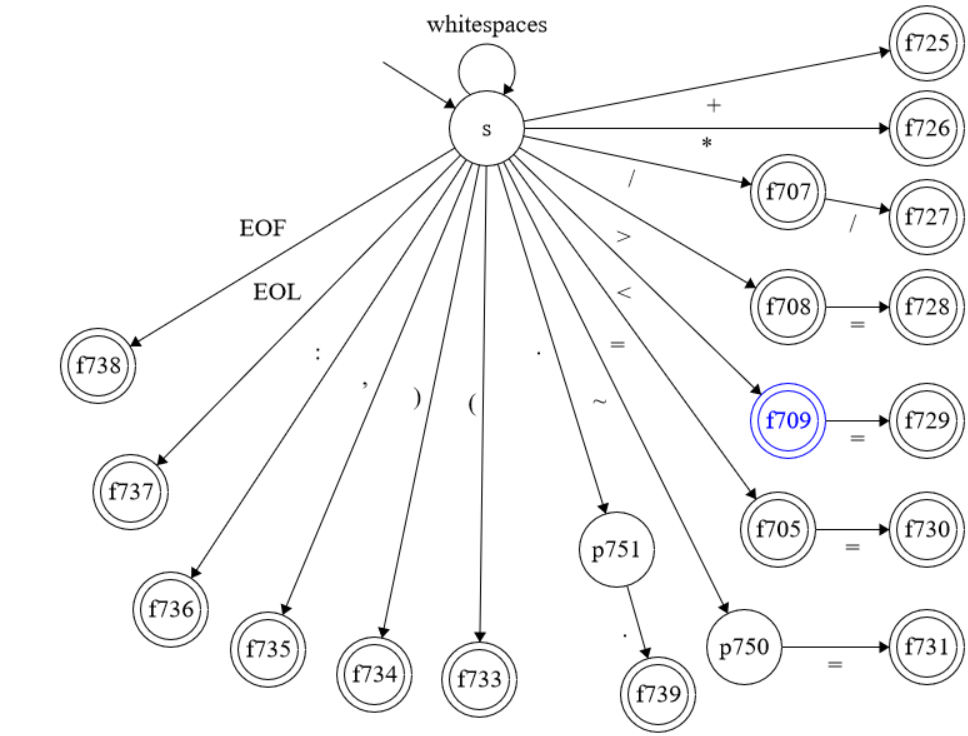
# Konecny automat stavov pre lexikalny analyzator

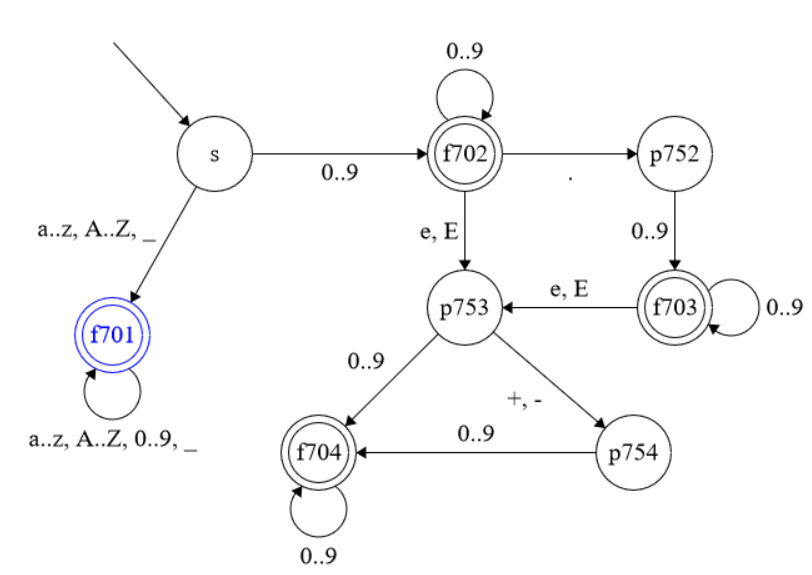
fsm beznych zakladnych stavov

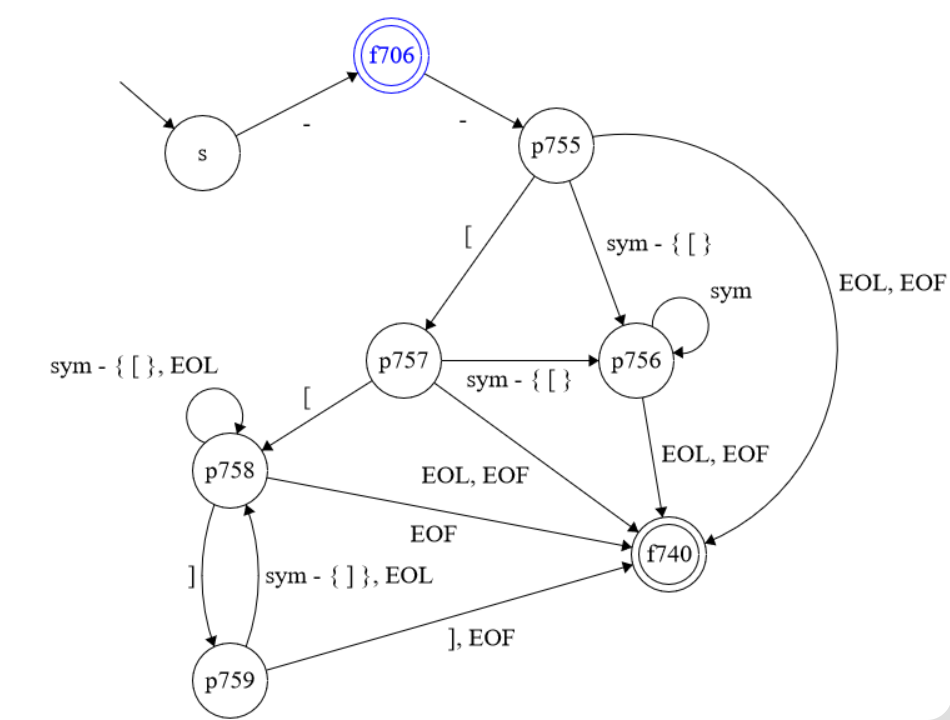
Uprava – prechod pomocou # do stavu f732

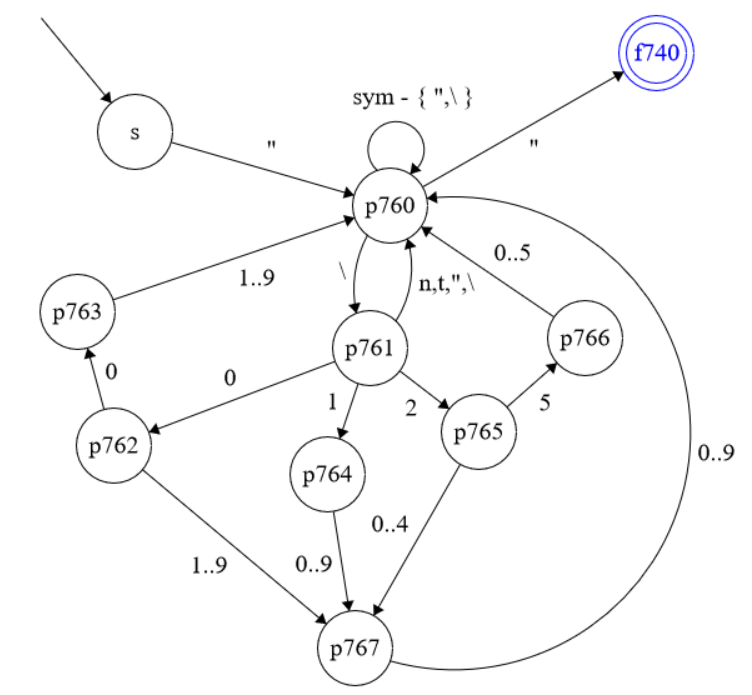
Uprava – whitespaces sa nahradia ako \t, ‘ ‘ \n

Uprava – odstranenie EOL



fsm keywords, IDs, data\_types, numbers

fsm minus, komentare

fsm stringy

**LL – GRAMATIKA**

1. <prol> → REQUIRE "ifj21" <prog>
2. <prog> → GLOBAL ID : FUNCTION ( <types> ) <return\_func\_var> <prog>
3. <prog> → FUNCTION ID ( <params> ) <return\_func\_var> <statement> END <prog>
4. <prog> → ID ( <expression> ) <prog>
5. <prog> → EOF
6. <return\_func\_var> → : <type> <type\_n>
7. <return\_func\_var> → ε
8. <params> → ID : <type> <param\_n>
9. <params> → ε
10. <param\_n> → , ID : <type> <param\_n>
11. <param\_n> → ε
12. <statement> → ε
13. <statement> → LOCAL ID : <type> <assign\_value> <statement>
14. <statement> → RETURN <expression> <statement>
15. <statement> → ID <assig\_func> <statement>
16. <statement> → IF <expression> THEN <statement> ELSE <statement> END <statement>
17. <statement> → WHILE <expression> DO <statement> END <statement>
18. <assign\_value> → ε
19. <assign\_value> → = <expression>
20. <id\_n> → ε
21. <id\_n> → , ID <id\_n>
22. <types> → ε
23. <types> → <type> <type\_n>
24. <type\_n> → ε
25. <type\_n> → , <type> <type\_n>
26. <type> → INTEGER
27. <type> → STRING
28. <type> → NUMBER
29. <type> → NIL
30. <assig\_func> → <id\_n> = <expression>
31. <assig\_func> → ( <expression> )