

Airplane Language

APS - Logcomp

"Aviões são muito mais
do que meios de
transporte, são
máquinas que levam
sonhos aos seus
destinos"

MOTIVAÇÃO

Aviation Language (.fly) foi criada por um apaixonado por aviões com intuito de simular o FMGS (flight management guidance system) de maneira simplificada. Essa linguagem tem tipagem e verificação para waypoints.

O que é FMGS?

FMGS (flight management guidance system) ou FMC (flight management computer) é o computador do avião responsável pela navegação vertical e lateral das aeronaves modernas. É nesse computador que os pilotos configuram as rotas, procedimentos de aproximação e até mesmo dados de performance durante decolagem, subida, cruzeiro, descida e pouso. É nesse computador que é possível configurar um voo todo, com precisão e segurança.

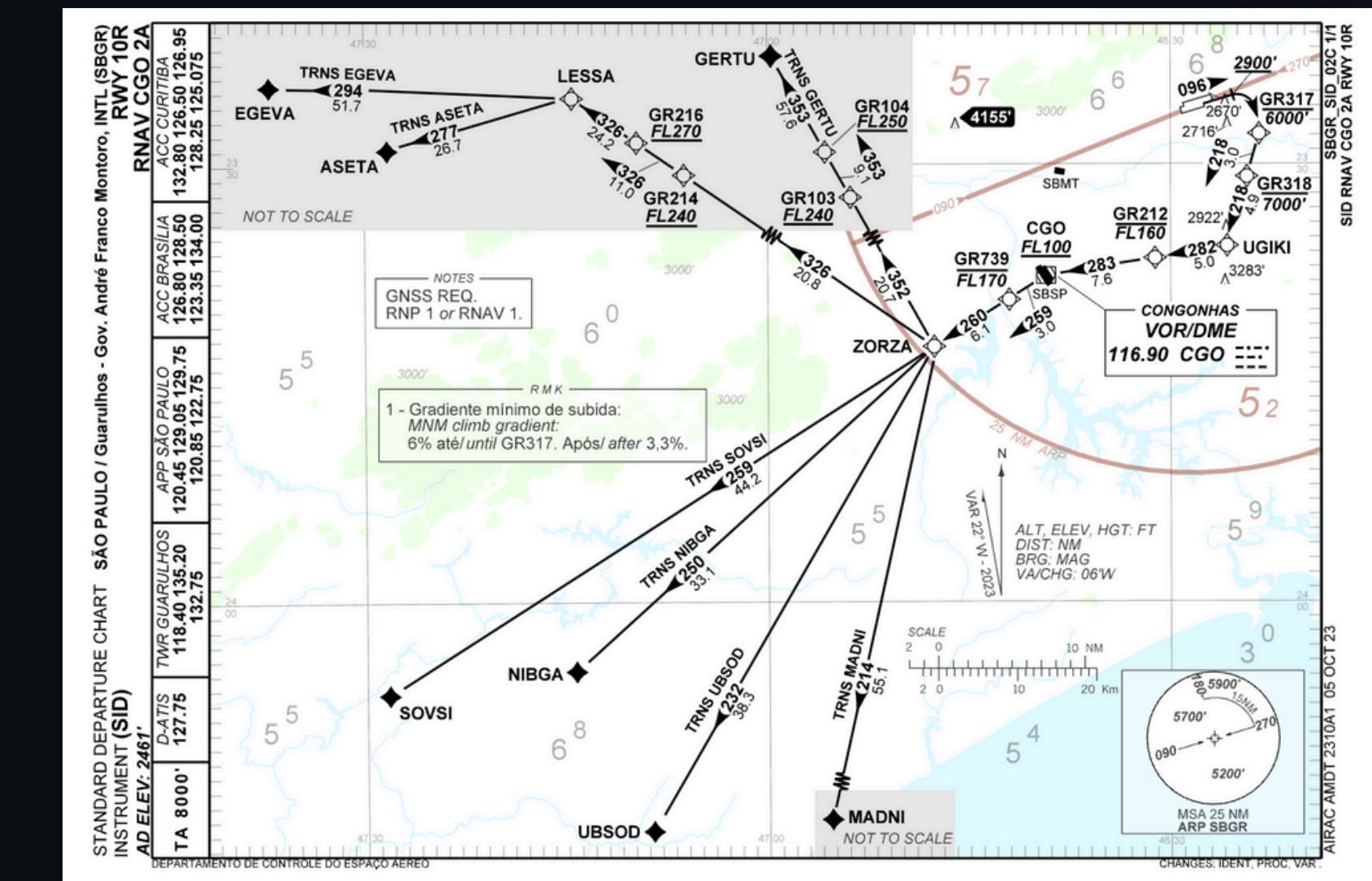


O que é AIRAC?

AIRAC (Aeronautical Information Regulation and Control) é um sistema de regulação e controle de informações aeronáuticas utilizado internacionalmente. Ele foi criado pela Organização de Aviação Civil Internacional (ICAO) para garantir a segurança e a eficiência das operações aéreas.

No compilador da linguagem os fixos do ciclo 2404 do airac são carregados para efetuar a validação de waypoints.

Na imagem ao lado há exemplo de uma carta de saída (SID) do aeroporto de Guarulhos, alguns fixos válidos são SOVSI e UBSOD, por exemplo.



EBNF da linguagem

```
○ ○ ○ EBNF

1 BLOCK = { STATEMENT };
2
3 STATEMENT = ( | ASSIGNMENT | WHILE | FUNCTION | IF | COMMENT | RETURN );
4
5 ASSIGNMENT = { IDENTIFIER, "→", TYPE, "=", EXPRESSION };
6
7 PARAMETER = { IDENTIFIER, "→", TYPE, {"", IDENTIFIER, "→", TYPE } } ;
8
9 TYPE = ( "int" | "float" | "string" | "waypoint" | "flight_level" );
10
11 WHILE = "while", "(", RELEXPRESSION, ")" , "{", STATEMENT, "}";
12
13 FUNCTION = "function", IDENTIFIER, "(", [PARAMETER], ")" , STATEMENT ;
14
15 IF = "if", "(", EXPRESSION, ")" , STATEMENT ;
16
17 PRINT = "print", "(", EXPRESSION, ")" ;
18
19 RELEXPRESSION = EXPRESSION, ( "==" | "!=" | "<" | ">" | "≤" | "≥" ), EXPRESSION;
20
21 EXPRESSION = TERM, { ("+" | "-"), TERM } ;
22
23 TERM = FACTOR, { ("*" | "/"), FACTOR } ;
24
25 MATHFUNC = ( "sin" | "cos" | "tan" | "asin" | "acos" | "atan" | "sqrt" | "pow" | "log"
| "exp" | "abs" | "ceil" | "floor" | "round" );
26
27 FACTOR = (( "+" | "-" | "!" ), FACTOR ) | NUMBER | STRING | "(", REEXPR, ")" |
IDENTIFIER, [ "(", REEXPR, {"", REEXPR}, ")" ] | ( "MATHFUNC", "(", REEXPR, {"", REEXPR}, ")" );
28
29 IDENTIFIER = LETTER, { LETTER | DIGIT | "_" } ;
30
31 NUMBER = DIGIT, { DIGIT } ;
32
33 RETURN = "return", RELEXPRESSION;
34
35 DIGIT = ( "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ) ;
36
37 LETTER = ( "a" | "..." | "z" | "A" | "..." | "Z" ) ;
```

Etapas do compilador (feito em python)



Pré pro

Remove comentários e também carrega o AIRAC 2104 em memória



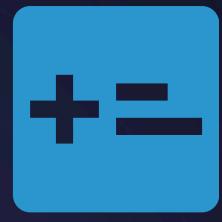
Tokenizer

Valida os tokens do programa e gera-os para a próxima etapa



Parser

Analisa a ordem dos tokens e gera a AST



Nodes

Evaluate dos nodes gerados pelo Parser

Exemplos

Os exemplos podem ser encontrados no repositório
<https://github.com/andrebrito16/airplanelanguage> (arquivos .fly) e no
README do repositório.

CURIOSIDADES

Sou um grande apaixonado por aviões e essa foto aí é de um Airbus A320 (study level) em um simulador

