

Lab7:

<https://github.com/davidalexandru1370/LabLFTC>

on branch Lab7

Calcul FIRST

1. Initializare. Pentru fiecare neterminal A al gramaticii se formează mulțimea $F_0(A)$ după cum urmează: se identifică acele producții pentru neterminalul A , a căror parte dreaptă începe cu simbol terminal. Simbolurile terminale care se află pe prima poziție în partea dreaptă a produțiilor lui A se adaugă în mulțimea $F_0(A)$. În cazul în care există și epsilon producție pentru A ($A \rightarrow \epsilon$), mulțimea $F_0(A)$ îl va conține și pe ϵ .

2. Iteratii. În iteratia i se calculează $F_i(A)$ pentru fiecare neterminal A al gramaticii. (Iteratia $i-1$ fiind încheiată, cunoaștem valorile lui $F_{i-1}(B)$, pentru fiecare neterminal B .) Mai întâi, în mulțimea $F_i(A)$ se adaugă toate elementele din $F_{i-1}(A)$. Apoi se identifică toate părțile drepte ale produțiilor lui A . Fie α o astfel de parte dreaptă, formată, în ordine, din simbolurile X_1, X_2, \dots, X_n . Pentru fiecare $X_j, j=1, n$, cunoaștem $F_{i-1}(X_j)$. Se concatenează toate aceste mulțimi $F_{i-1}(X_j)$, în ordine, și din secvențele rezultate în urma concatenării se consideră doar primul simbol. Aceste prime simboluri se adaugă în mulțimea $F_i(A)$. Dacă cel puțin o mulțime $F_{i-1}(X_j)$ este vidă, calculul nu se mai efectuează.

Iteratiile continuă până în momentul în care $F_i(A) = F_{i-1}(A)$ pentru toate neterminalele A . (două coloane consecutive ale tabelului sunt egale). Ultima coloană găsită (în iteratia i), da valorile funcției FIRST.

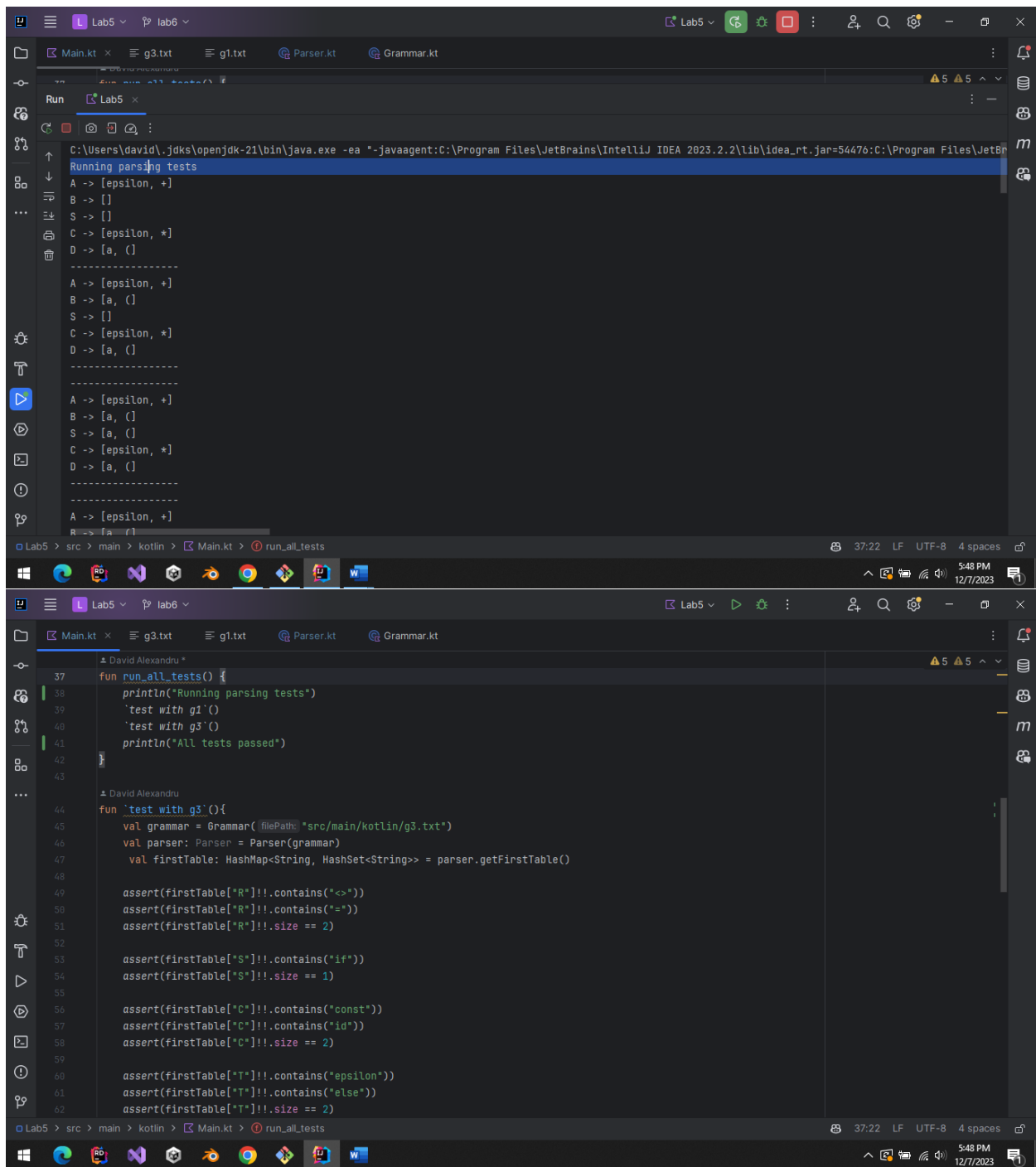
Calcul FOLLOW

1. Initializare. Pentru fiecare neterminal A al gramaticii, mai puțin simbolul de start, se inițializează mulțimea $L_0(A)$ cu mulțimea vidă. Pentru simbolul de start S , $L_0(S)$ va conține doar pe epsilon.

2. Iteratii. În iteratia i calculăm $L_i(A)$ pentru fiecare neterminal A . Se adaugă în $L_i(A)$ toate simbolurile din $L_{i-1}(A)$. Apoi, se identifică toate producțiile gramaticii care îl conțin pe A în partea dreaptă. Fie β secvența care îi urmează lui A într-o astfel de parte dreaptă. Se calculează $FIRST(\beta)$ și se adaugă simbolurile diferite de epsilon din $FIRST(\beta)$ în $L_i(A)$. Dacă $FIRST(\beta)$ îl conține pe epsilon (sau dacă β e chiar epsilon) nu se adaugă epsilon în $L_i(A)$, ci se adaugă toate simbolurile din $L_{i-1}(B)$, unde B e neterminalul din partea stângă a producției curente.

Se continuă iteratiile până când două coloane consecutive sunt în întregime egale. Ultima coloană găsită da valorile lui FOLLOW.

Preluat din referințele de la seminar



```

R -> [const, id]
S -> [epsilon]
C -> [then]
T -> [epsilon]
E -> [<>, =]
I -> [epsilon, else]
-----
R -> [const, id]
S -> [epsilon]
C -> [then]
T -> [epsilon]
E -> [epsilon, <>, else, then, =]
I -> [epsilon, else]
-----
R -> [const, id]
S -> [epsilon]
C -> [then]
T -> [epsilon]
E -> [epsilon, <>, else, then, =]
I -> [epsilon, else]
-----
All tests passed

```

Calcul output table:

Modul în care va functiona este dupa modelul unui automat push—down având ca date de intrare: gramatica limbajului tabelul de analiză LL(1) și secventa de analizat. Automatul push-down are o bandă de intrare, numită stivă de intrare, O stivă de lucru, unitatea centrală al cărei comportament -este definit prin tranziții și o bandă de ieșire

Tabelul de analiză LL(1) trebuie să cuprindă acțiunile posibile luând în considerare simbolul curent $X \in N \cup \Sigma$ și predicțiile posibile $u \in \Sigma$. Astfel vom avea un tabel bidimensional, care va conține câte o linie pentru fiecare simbol din gramatică (terminale și neterminale) și câte o coloană pentru fiecare predicție posibila. În plus, se adaugă un caracter special, de obicei notat '\$' al cărui scop este să marcheze sfârșitul secvenței și căruia i se alocă o linie și o coloană în tabel. Efectul acestui simbol în faza de analiză propriu-zisă este de a elimina verificările de stivă goală. Regulile de completare a tabelului sunt:

(Sursa: manual LFTC)