

Limbaje formale și tehnici de compilare

analiza de tipuri

Analiza de tipuri (AT) verifică dacă simbolurile, constantele și în general instrucțiunile sunt folosite în mod corect. De exemplu, o eroare care este depistată la AT este apelarea unei variabile ca și când ar fi fost o funcție. Toate regulile semantice pentru AT a limbajului Quick sunt descrise în comentarii la începutul fișierului „**Quick - analiza de tipuri.pdf**”.

Implementarea analizei de tipuri pentru limbajul Quick

În limbajul Quick analiza de tipuri are loc simultan cu analiza sintactică. Pentru aceasta, în unele reguli sintactice se va insera cod care implementează acțiunile necesare analizei de tipuri în acel punct al analizorului sintactic. Toate aceste fragmente de cod sunt date în fișierul „**Quick - analiza de tipuri.pdf**”, între acolade și folosind culoarea albastru.

Secvențele de cod pentru analiza de domeniu folosesc anumite funcții, care sunt declarate și implementate în fișierele „**at.h**” și „**at.c**”. Aceste fișiere vor trebui incluse în compilatorul propriu.

Fișierul „**at.c**” trebuie să aibă acces la definițiile de tipuri (**TYPE_***) și la funcțiile de la analiza de domeniu. Din acest motiv, la începutul lui a fost inclus antetul „**main.h**”, care este un antet global, care cuprinde toate definițiile din proiect. În funcție de implementările proprii, puteți folosi alte soluții care să asigure în „**at.c**” accesul la definițiile necesare.

Pentru a se putea implementa AT, a fost nevoie să se scrie unele reguli din gramatica Quick într-un format echivalent. Este vorba de regulile care au componente opționale (ex: „**exprAssign ::= (ID ASSIGN)? exprComp**”), în așa fel încât să se poată include cod atât dacă există componenta opțională, cât și dacă nu există. De obicei această rescriere s-a făcut prin dublarea fragmentului respectiv folosind alternative, astfel încât să fie explicite ambele variante, de existență și de non-existență a componentei opționale. În tabelul de mai jos sunt toate fragmentele de reguli care au fost rescrise („**epsilon**” înseamnă „**nimic**”, adică ramura se îndeplinește fără a se consuma nimic). Fragmentele modificate deja au fost incluse în fișierul „**Quick - analiza de tipuri.pdf**”, astfel încât trebuie doar să fiți atenți cum modificați analizorul sintactic propriu.

Fragment original	Fragment modificat
ID (LPAR (expr (COMMA expr) *)? RPAR)?	ID (LPAR (expr (COMMA expr) *)? RPAR epsilon)
(SUB NOT)? factor	SUB factor NOT factor factor
(ID ASSIGN)? exprComp	ID ASSIGN exprComp exprComp

Evaluarea activității

Evaluarea acestei activități se va face conform următoarei proceduri:

1. Se va folosi ca intrare în compilator fișierul exemplu din laborator. În varianta originală nu trebuie să apară niciun mesaj de eroare.
2. Se vor introduce în exemplu unele erori care să contrazică regulile semantice de la începutul fișierului „**Quick - analiza de tipuri.pdf**”, iar compilatorul va trebui să genereze mesajele de eroare specifice. De exemplu, dacă la sfârșitul exemplului se scrie „**x=1;**”, se va emite mesajul de eroare „*eroare in linia 20: identificador necunoscut: x*”.
3. Studentul va fi necesar să știe să explice cum funcționează codul nou adăugat, inclusiv cel din fișierul „**at.c**”