Implementácia prekladača imperatívneho jazyka IFJ21

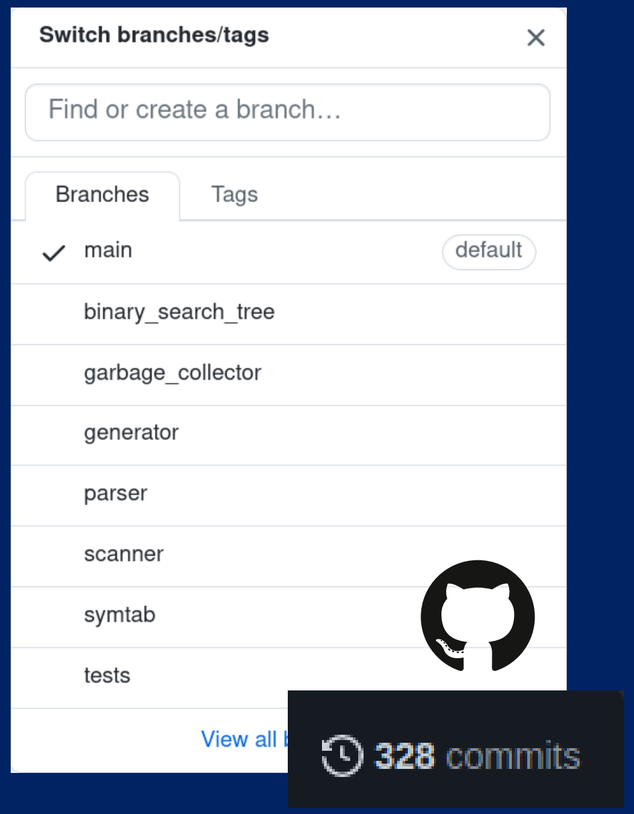
JIŘINA FRÝBORTOVÁ, JANA KOVÁČIKOVÁ, ALEXANDER OKRUCKÝ, PATRIK SKALOŠ

TÍM 117, VARIANTA I

ORGANIZÁCIA V TÍME

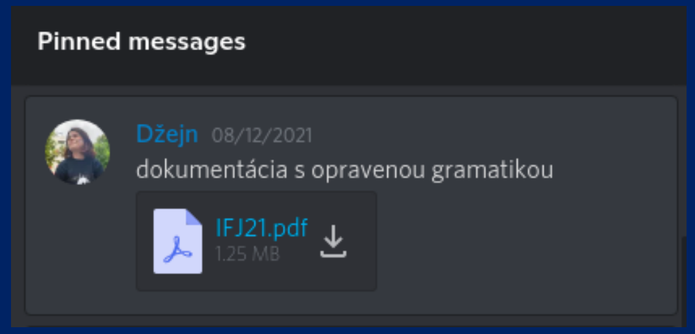
GitHub

- verzovací nástroj
- rozdelenie do vetví



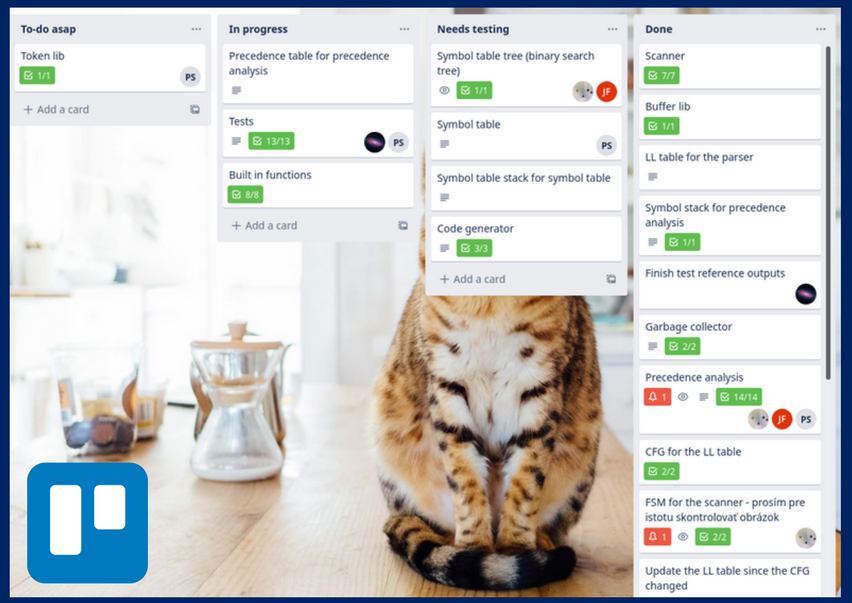
Discord

- komunikačný kanál
- miestnosti na písanie, dôležité termíny



Trello

- to-do list
- úlohy sú v stĺpcoch podľa ich stavu



REALIZÁCIA

- jednoprechodová analýza zdrojového kódu
- generovanie bez medzikódu
- v prípade nájdenia chyby sa preklad kódu ihneď a bez zotavenia ukončí
- výpis chybovej hlášky a približného čísla riadku chyby na štandardný chybový výstup

SKENER

- konečný automat
- v prípade nájdenia chyby vracia volajúcej funkcii chybový kód
- štruktúra tokenu obsahuje informáciu o type tokenu a jeho obsahu
- identifikátory a kľúčové slová vraciame ako jeden typ tokenu

TABUĽKA SYMBOLOV

- zásobník klasických binárnych vyhľadávacích stromov
- každý prvok stromu reprezentuje jeden identifikátor
- pred behom prekladača sa do tabuľky vkladajú vstavané funkcie

PRECEDENČNÁ ANALÝZA

- volaná parserom na spracovanie výrazu
- využíva zásobník symbolov
- pri aplikácií jedného z pravidiel generuje kód
- ak nie je možné aplikovať žiadne z pravidiel, nastala chyba
- algoritmus prevzatý z prednášok IFJ

PARSER

- zvolili sme rekurzívny zostup
- keď narazí na výraz, zavolá precedenčnú analýzu
- všetky premenné deklarujeme až na konci vygenerovanej funkcie
- kopíruje našu gramatiku

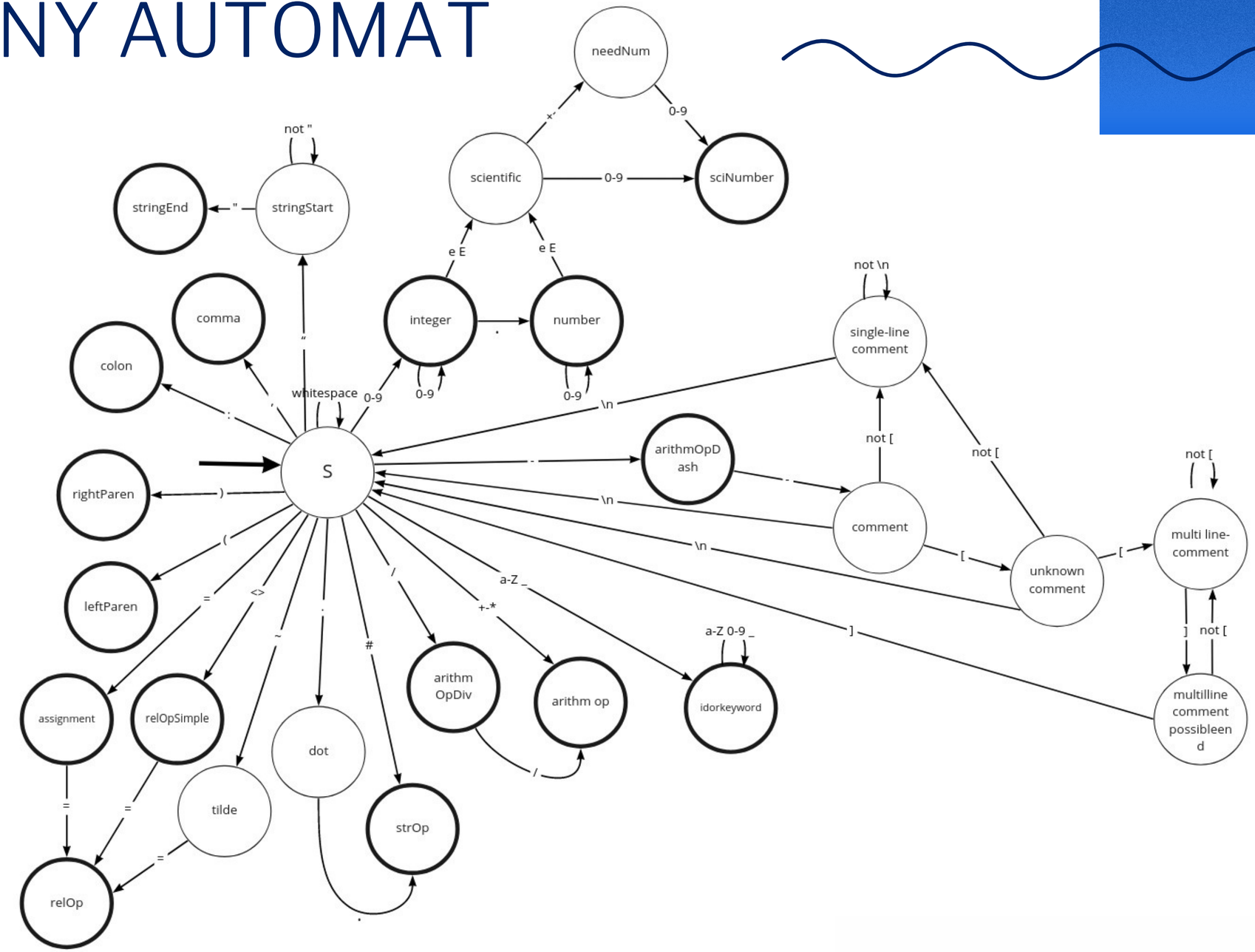
GENERÁTOR CIEĽOVÉHO KÓDU

- generujeme bez medzikódu
- z parseru voláme funkcie na generovanie cieľového kódu IFJcode21

VYUŽITÉ DÁTOVÉ ŠTRUKTÚRY

- Nafukovacie polia:
  - znakov
  - reťazcov
  - čísel
- Zásobník:
  - binárnych stromov
  - symbolov
- Binárny strom
- Obojsmerne viazaný zoznam

"Veľa vecí sme fixli, hlavne tie čo fungujú, to čo nefunguje treba opraviť."

- Alexander Okrucký

KONEČNÝ AUTOMAT