**Vysoké učení technické v Brně**

**Fakulta informačních technologií**

Dokumentace projektu do předmětů IFJ a IAL

**Implementace překladače jazyka IFJ22**

Tým xzavad20 – varianta TRP

Adam Světlík: xsvetl07 – 25%

Marek Procházka: xproch0o – 25%

Lukáš Zavadil (vedoucí): xzavad20 – 25%

David Sklenář: xsklen14 – 25% 2022

# Obsah

Obsah se vkládá až po napsání celé práce pomocí menu Odkazy / Obsah. Při jakékoliv změně nadpisů v práci je třeba obsah aktualizovat – pravé tlačítko / Aktualizovat pole / celá tabulka.

[Obsah 2](#_Toc121273817)

[Rozdělení práce v týmu 3](#_Toc121273818)

[Implementace 4](#_Toc121273819)

[Lexikální analýza 4](#_Toc121273820)

[Seznam obrázků 5](#_Toc121273821)

# Rozdělení práce v týmu

Rozdělení částí projektu:

* Adam Světlík – návrh a implementace tabulky symbolů, návrh a implementace generace kódu
* Marek Procházka – správa společného repozitáře, gitflow management, automatizace testů, návrh a implementace souboru pro práci s dynamickými řetězci, implementace syntaktického analyzátoru
* Lukáš Zavadil – jako vedoucí řídil práci na projektu, rozdělení práce mezi členy týmu, návrh a implementace precedenční analýzy, návrh syntaktického analyzátoru, návrh gramatiky
* David Sklenář – návrh a implementace lexikálního analyzátoru, implementace části generace kódu, tvorba dokumentace

Zároveň v průběhu práce na projektu každý člen týmu psal testy pro své části a kompilátor jsme jako celek na konci společně testovali jak našimi testy, tak i testy které poskytnuli naši spolužáci.

Ke sdílení a organizaci práce jsme používali verzovací systém Git s repozitářem na GitHubu.

TODO: lukeho prezentace podojit, sessiony podojit, howto flow pracovní, bydlíme spolem, ze jsme vseci debuggovali spolu

# Implementace

## Lexikální analýza

Lexikální analyzátor je implementovaný v souboru scanner.c a v hlavičkovém souboru scanner.h. Tato část kompilátoru má na starost zpracování vstupního programu v jazyce IFJ22, který v našem případě přichází na stdin, a jeho rozložení na tokeny, které pak následně předává ke zpracování dalším částem kompilátoru.

Lexikální analyzátor je implementovaný na základě konečného automatu. Jeho návrh probíhal postupně procházením zadání, vzorových zdrojových kódů a tvořením různých stavů automatu. Chybné stavy byly průběžně odchytávány, hlavně při implementaci. Výsledkem je graf deterministického konečného automatu, který je obsažen v příloze 1.

V samotném scanneru je nejdůležitější funkce scan(\*token). Ta slouží pro komunikaci ostatních částí kompilátoru se scannerem. Při každém zavolání vrátí jeden přes parametr jeden token, který obsahuje typ aktuálního tokenu a popřípadě jeho hodnotu, pokud je u tokenu vyžadována. Zároveň vrací návratovou hodnotu, pomocí které se rozhoduje zda kompilace zatím probíhá v pořádku, či zda došlo k lexikální nebo syntaktické chybě. Pro své fungování hlavní funkce scan používá několik pomocných funkcí, například pro zpracování řetězců, klíčových slov, celočíselných a desetinných literálů. Předávání z těchto funkcí stále probíhá do předpřipravené struktury.

## Tabulka symbolů

TODO – nezapomenout zmínit pepeho a zjistit kde ukradl hashovací funkci ideálně (ale kdyžtak bude asi pepe stačit)

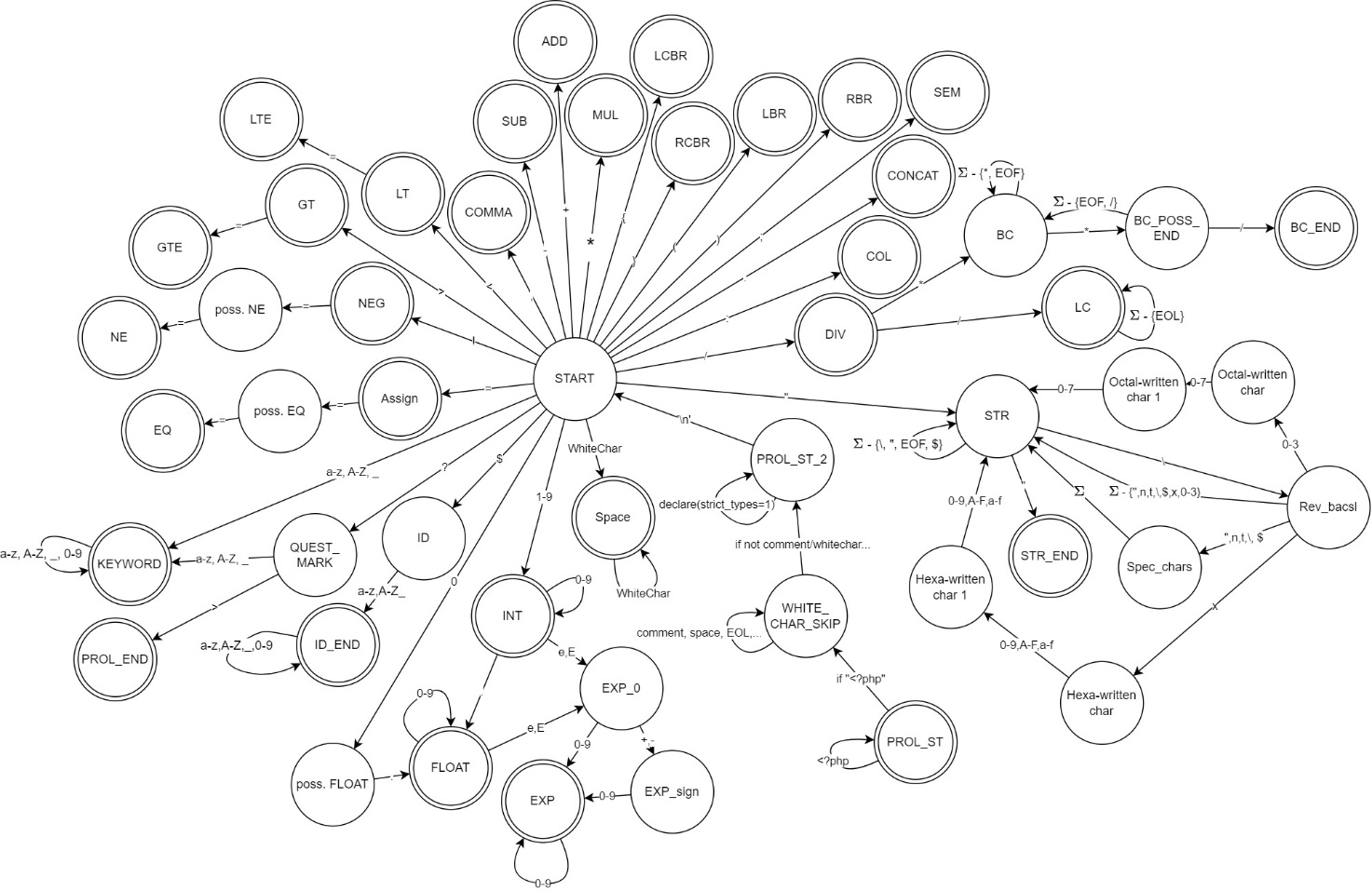
# Seznam obrázků

Zde se vkládá jí seznamy obrázků a tabulek. Vložíte to v menu Odkazy / Vložit seznam obrázků, kde vyberete v Obecných / Popisek titulu Obrázek č. a potom Tabulka č.

**Nenalezena položka seznamu obrázků.**

**Nenalezena položka seznamu obrázků.**

# Přílohy

 Příloha 1 – graf konečného automatu lexikálního analyzátoru