Systemnahes Programmieren

Dokumentation

Teil 1

WS 2016/17

Boris Bopp

Maksim Sunjajkin

# Vorwort

Ziel dieser Übung war es die Funktion eines Scanners als auch die eines Compilers kennenzulernen und die vorhandenen C/C++ Kenntnisse zu vertiefen.

Ein Compiler ist ein Programm das eine gegebenen Sprache in eine andere übersetzt.

Zum Beispiel: C/C++ -> Maschinensprache

Im ersten Teil unserer Übung beschäftigen wir uns mit der Lexikalischen Analyse. Das heißt mit der Zerlegung des Quellcodes in seine Bestandteile und einer darauffolgenden Erstellung einer Zwischendarstellung.

Inhaltsverzeichnis

[Vorwort 2](#_Toc480297527)

[Architektur 4](#_Toc480297528)

[Verzeichnisse und Dateien 5](#_Toc480297529)

[Code Dokumentation 6](#_Toc480297530)

[Buffer 6](#_Toc480297531)

[Scanner 6](#_Toc480297532)

[Automat 7](#_Toc480297533)

[Symboltabele 8](#_Toc480297534)

[Aufgetretene Probleme und Lösungen 9](#_Toc480297535)

# Architektur

Die im ersten Teil zu bearbeitenden Teile des Compilers sind folgende:

* Buffer: Zuständig für die Eingabe, der Puffer besitz eine maximallänge und Speichert einen Teil der Eingabe zwischen.
* Automat: Der Automat ist bei der Token Erzeugung für die Erkennung des Typen zuständig, er besitzt einen Zustand der je nach eingelesenen Zeichen sich ändern kann.
* Scanner: Der Scanner verbindet Puffer, Automat und Symboltabelle. Er nimmt solange Zeichen vom Puffer entgegen und gibt sie an den Automaten weiter bis er ein gültiges Token erstellen kann, die daraus resultierenden Zustände speichert er in der Symboltabelle

* Tokens: Mit Tokens werden die Grundsymbole bezeichnet.
* Symboltabelle: In der Symboltabelle werden die Informationen die in Verbindung zu dem Token stehen gespeichert

# Verzeichnisse und Dateien

|-- Automat/

| |-- includes/

| |-- src/

| |-- makefile

|-- Buffer/

| |-- includes/

| |-- src/

| |-- makefile

|-- Common/

| |-- includes/

| |-- src/

| |-- makefile

|-- Scanner/

| |-- includes/

| |-- src/

| |-- makefile

|-- Symboltable/

| |-- includes/

| |-- src/

| |-- makefile

|-- makefile

Das Projekt wurde in 4 Teilprojekte unterteilt: Automat, Buffer, Scanner und Symboltable.

Jedes Teilprojekt enthält sowie ein eigenes makefile als auch eine main-Klasse um unabhängig die Funktionalität testen zu können.

Makefiles werden benötigt um das Compilieren und Linken automatisiert ausführen zu können.

Im Verzeichnis Common befinden sich selbst erstellte Libraries die in die Teilprojekte mit einbezogen werden.

Um das Gesamtprojekt zu Compilieren wird das makefile der obersten Ebene ausgeführt.

# Code Dokumentation

## Buffer

Wenn nicht anders vorgegeben wird ein Buffer mit einer Größe von 1024 Zeichen erstellt. Dieser wird mithilfe der Library *fstream* befüllt. Es werden jedoch jeweils nur [Buffergröße - 1] Zeichen kopiert damit man im Falle eines Dateiendes bei [Buffergröße - 1]ten Zeichen noch ein Eof Indikator einfügen kann.

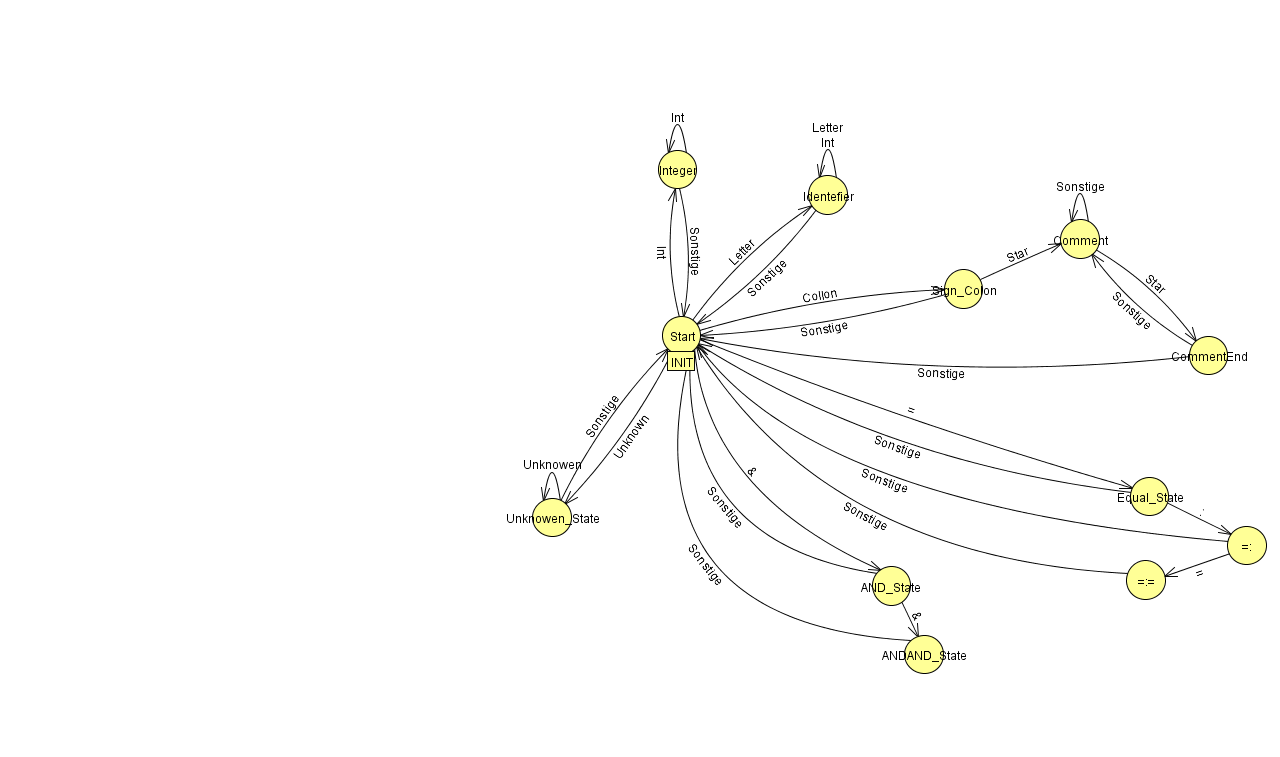
Der Buffer wird zusätzlich als Node in einer Verkettetenliste gespeichert was uns erlaubt ein Zeichen aus einem vorherigen Bufferblock zu lesen.

## Scanner

Der Scanner liest mithilfe des Buffers Zeichen ein und gibt sie an den Automaten weiter. Der Automat bestimmt ihren Typ und gibt ihn an den Scanner. Mit dieser Information kann der Scanner ein Token erstellen und ihn hinzufügen falls es diesen noch nicht gibt. Sollte dieser Token schon existieren wird nur die Lexem Information(Wert, Position) hinzugefügt.

## Automat

Folgendes Bild zeigt die Zustände und Übergänge des Programmierten Automaten



## Symboltabele

## Aufgetretene Probleme und Lösungen

Zur ersten Abgabe hatte das Programm drei Fehler, diese wären:

1. Das Token =: wurde als Unknownen deklariert

Lösung: Im Automaten war ein Übergang falsch beschrieben -> Verbesserung am Automaten

1. Das EoF Token wurde nicht erstellt, wenn vor dem Ende der Datei keine Leerzeile bestand.

Lösung: Wir fügen nun eine Leerzeile ein sollte es zu Schluss keine geben.

1. Zeilen die während einem Kommentar eingelesen wurden, wurden nicht mitgezählt.

Lösung: Einen extra zustand für Kommentare bei dem auch die Zeilenmitgezählt werden.