

Bachelorprojekt i Oversætterkonstruktion

Kitty

Maj 2019

**Rapport fra gruppe GRUPPENUMMER:
STUDERENDE1, STUDERENDE2, STUDERENDE3,
STUDERENDE4**

1 Indledning

1.1 Afklaringer

Har I haft brug for at foretage nogle præciseringer i sprogdefinitionen? Forklar.

1.2 Begrænsninger

Har I bevidst lavet nogle begrænsninger i jeres version af sproget? Hvad skyldes det? Hvad er følgerne?

1.3 Udvidelser

Har I lavet nogle udvidelser i jeres version af sproget? Hvad var jeres motivation? Hvad er følgerne?

1.4 Implementationsstatus

Hvad er status af jeres implementation? Er alle de foreslåede funktionaliteter implementerede? Er de afprøvede? Virker de?

2 Parsning og abstrakte syntakstræer

2.1 Grammatikken

Angiv den fulde grammatik for jeres version af sproget f.eks. ved at liste jeres `bison` input med alle actions fjernet, eller også en anden pænere måde at angive en grammatik.

2.2 Brug af `flex` værktøjet

Kort introduktion, hvor man lige i en paragraf fortæller, hvad `flex` gør. Diskuter interessante dele af jeres `flex` implementation af scanneren. Hvilke tokens benytter I? Bruger I start conditions? Hvordan og hvorfor? Hvordan håndteres kommentarer og white space. Er der konstruktioner, der af en eller anden grund skal behandles specielt?

2.3 Brug af `bison` værktøjet

Diskuter interessante dele af jeres `bison` implementation af parseren. Hvordan fik I `bison` til at acceptere jeres grammatik? Hvad er jeres design af precedence, og hvordan har I realiseret det? Giv en oversigt over, hvor stærkt de forskellige operatorer binder. Er der konstruktioner, der af en eller anden grund skal behandles specielt?

2.4 Abstrakte syntakstræer

Angiv strukturen af jeres abstrakte syntakstræer, herunder specielt hvilke felter I har bestemt jer til at have i knuderne, og kort hvad de skal bruges til under hvilke faser.

2.5 Afsukring

Behandler I syntaktisk sukker under parsningen? Hvordan og hvorfor?

2.6 Udlugning

Udluger I uønskede syntakstræer? Hvordan og hvorfor?

2.7 Afprøvning

Har I afprøvet denne fase? Hvordan? Virker den?

3 Symboltabeller

3.1 Scoperegler

Angiv eventuelle afklaringer af scopereglerne i jeres sprog.

3.2 Symboldata

Beskriv indholdet af symboltabellens indgange.

3.3 Algoritme

Beskriv jeres algoritme til at opbygge symboltabeller og checke scoperegler.

3.4 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

4 Typecheck

4.1 Typer

Beskriv de typer, som jeres sprog understøtter.

4.2 Typeregler

Beskriv typereglerne i jeres sprog.

4.3 Algoritme

Beskriv jeres algoritme for at checke typeregler.

4.4 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

5 Ressourceberegning

5.1 Ressourcer

Beskriv de ressourcer, som I beregner. Er dette en separat fase?

5.2 Algoritme

Beskriv eventuelle algoritmer til at beregne ressourcer.

5.3 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

6 Kodegenerering

6.1 Strategi

Beskriv den overordnede strategi for kodegenerering til mellemform, samt jeres stack frame.

6.2 Kodeskabeloner

Beskriv kodeskabeloner for sprogkonstruktioner.

6.3 Algoritme

Beskriv jeres algoritme for at generere kode.

6.4 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

7 Faser før emit

Hvilke faser har I efter mellemformen er genereret og før emit?

7.1 Analyser

Hvilke analyser foretages, og hvad baserer de sig på?

7.2 Algoritmer

Beskriv jeres algoritmer til analyserne.

7.3 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

8 Emit

8.1 Eksempelkode

Angiv den komplette kode, der genereres, for et lille antal velvalgte og meget små eksempler; herunder `Factorial` fra hjemmesiden samt et eksempel, der involverer et array.

Dokumentér også gerne evt. ekstra-faciliteter.

8.2 Afprøvning

Har I afprøvet denne fase? Hvordan? virker den?

9 Konklusion

Evt. konklusion, samt dato og underskrifter.

A Kildetekster

I rækkefølge efter faser.