

Simple-Pascali programmide optimeerimine Haskellis andmevoanalyüsi teegiga Hoopl

Bakalaureusetöö

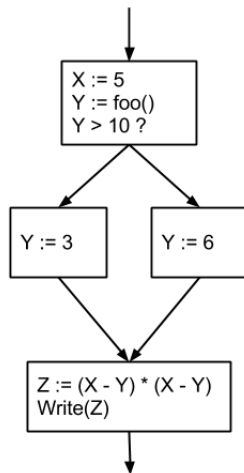
Tõnn Talvik
juhendaja: prof. Tarmo Uustalu

Tallinna Tehnikaülikool

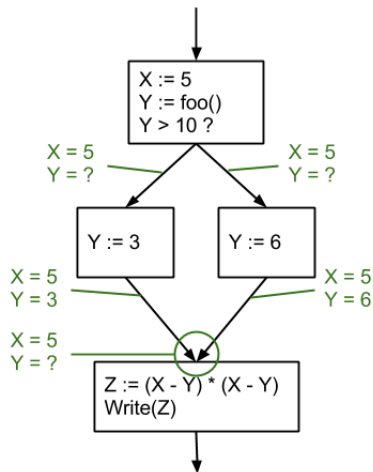
6. juuni 2013. a.

1. Teemast ja töö eesmärk
2. Tehtu ja tulemused
3. Mis oli huvitav

- ▶ vähendame korduseid & eemaldame liigse -> programmi täitmine kiirem
- ▶ andmevooanalüüs
 1. koostame juhtvoo graafi



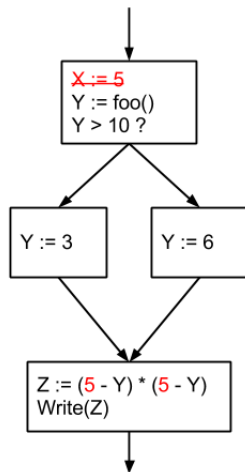
- ▶ vähendame korduseid & eemaldame liigse -> programmi täitmine kiirem
- ▶ andmevooanalüüs
 1. koostame juhtvoo graafi
 2. uurime andmete levikut graafi servades



Teemast

Optimeerimine, andmevooanalüüs

- ▶ vähendame korduseid & eemaldame liigse -> programmi täitmine kiirem
- ▶ andmevooanalüüs
 1. koostame juhtvoo graafi
 2. uurime andmete levikut graafi servades
 3. saadud teadmiste põhjal muudame programmi

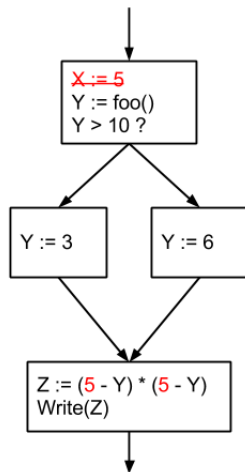


Teemast

Optimeerimine, andmevooanalüüs

- ▶ vähendame korduseid & eemaldame liigse -> programmi täitmine kiirem
- ▶ andmevooanalüüs
 1. koostame juhtvoo graafi
 2. uurime andmete levikut graafi servades
 3. saadud teadmiste põhjal muudame programmi

edasises veel näiteid



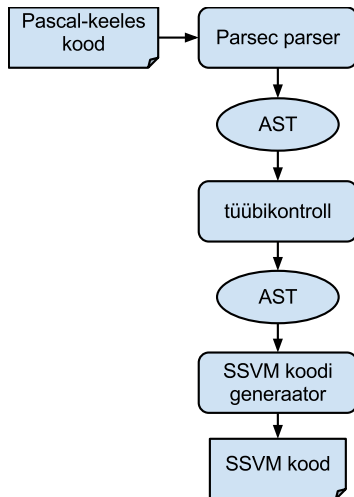
- ▶ funktsionaalkeel
- ▶ tööriistad
 - ▶ Glasgow Haskell Compiler (GHC)
 - ▶ Hoopl = andmevooanalüüsi teek
 - ▶ autorid: N. Ramsey, J. Dias, S. P. Jones
 - ▶ 2010
 - ▶ Parsec = parser
 - ▶ Simple-pascal...



Teemast

Simple-Pascal

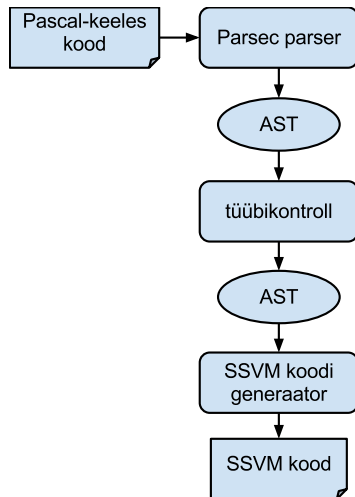
- ▶ Simple-Pascal = kompilaator, autor Ilya V. Portnov
- ▶ sisendiks lihtsustatud Pascal
- ▶ väljundiks Simple-Stacked-Virtual-Machine käsud



Teemast

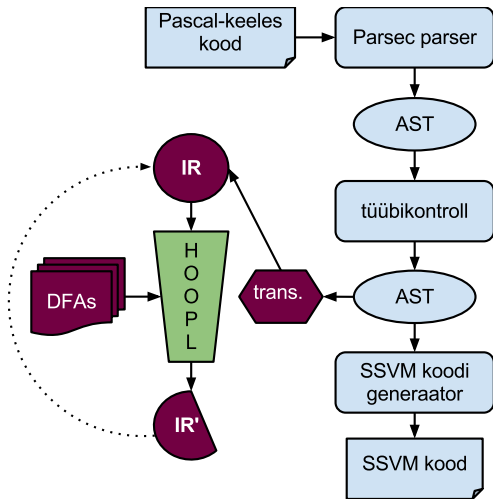
Simple-Pascal

- ▶ Simple-Pascal = kompilaator, autor Ilya V. Portnov
- ▶ sisendiks lihtsustatud Pascal
- ▶ väljundiks Simple-Stacked-Virtual-Machine käsud
- ▶ **optimeerimine puudub**



Lisada Simple-Pascalile
optimeerimine kasutades
Hoopl-teeki :

- ▶ vaheesituse (IR) loomine
- ▶ transleerimine AST -> IR
- ▶ andmevooanalüüsid (DFAs)



Tulemused

Teostatud andmevooanalüüsid

- ▶ konstantide voltimine ja levitamine
- ▶ koopiate levitamine
- ▶ ühiste alamavaldiste eemaldamine
- ▶ surnud omistamiste eemaldamine

Tulemused

Konstantide voltimine ja levitamine

- ▶ konstantidest koosnevad alamavaldised saab arvutada kompileerimise ajal
- ▶ $3 + 5$
- ▶ \rightarrow
- ▶ 8

Tulemused

Koopiate levitamine

- ▶ $y = x$
- ▶ $z = 3 + y$
- ▶ \longrightarrow
- ▶ $z = 3 + x$

Tulemused

Ühiste alamavaldiste eemaldamine

▶ $(a + b) - (a + b)/4$

▶ \longrightarrow

▶ $t = a + b$

▶ $t - t / 4$

Tulemused

Surnud omistamiste eemaldamine

- ▶ `x = 5`
- ▶ `x = foo()`
- ▶ `bar(x)`
- ▶ `—>`
- ▶ `x = foo()`
- ▶ `bar(x)`

Huvitavat

klassikalised probleemid - lahendused

- ▶ kompilaatori „juppideks võtmine“
 - ▶ grammatika, BNF
 - ▶ parsimine
 - ▶ optimeerimine: andmevooanalüüs
 - ▶ masinkäskude genereerimine
- ▶ mitmekülgsus => **tugev põhi**



„Compilers: Principles, Techniques, and Tools“ kaanepilt

<http://www.amazon.com/Compilers-Principles-Techniques-Alfred-Aho/dp/0201100886>

- ▶ monaadid & monaaditeisendajad
 - ▶ üldlevinud: Reader, State, IO
 - ▶ Hoopli: SimpleUniqueMonad, FuelMonad
 - ▶ isetehtu: TranslationMonad
- ▶ Haskellis kasutamine
 - ▶ silumine (võrreldes imperatiivkeeltega)
 - ▶ keele täiendused



GADTs, TypeFamilies,
RankNTypes,
ViewPatterns,
TypeOperators,
GeneralizedNewtypeDeriving,
ScopedTypeVariables

Täna tähelepanu eest!