

Synopsis - Modules in Futhark

vfr988

March 15, 2016

Problemformulering

Er det muligt at udvide Futhark med moduler, signaturer og funktorer, med SMLs modulsystem som inspiration [MLmodules]?

Begrundelse

SMLs modulsystem tillader definitionen af abstrakte strukturer, der konkretiseres gennem et funktor-kald med en type. Ved compile-time optimeres kald til modulerne væk, og efterlader almindelig kode.

Da Futhark oversættes til kode med en meget høj grad af parallelisering, vil et modulsystem give Futhark-udvikleren mulighed for at benytte sig af polymorfi, uden det går ud over det oversatte programs endelige køretid.

Muligheden for at inkludere eksterne moduler i et program betyder, at en programmør kan skrive et bibliotek af moduler, og siden hen inkludere disse moduler i sine programmer eksempelvis kan han skrive moduler til matrixregning, eller listefunktioner.

Succeskriterierne for projektet er følgende:

- **At Futharks modul-system oversættes til maskinkode korrekt**

Hvorvidt dette lykkes, vil primært blive målt på, hvorvidt Futhark succesfuldt udfører testprogrammer, der udnytter modulsystemet.

Der vil i bachelorrapporten også blive ført bevis for, at værdier der tilgås gennem modulsystemet, evaluerer på samme måde som værdier der tilgås i Futhark-kode, der ikke benytter modulsystemet. Såfremt dette er beviseligt, er succeskriteriet opfyldt.

- **At et Futhark-program der benytter modulsystemet, performer ligeså godt som et tilsvarende program, implementeret uden anvendelse af moduler**

Dette kriterie måles ved hjælp af måling af CPU-tid brugt af sammenlignelige Futhark-programmer, hvoraf kun det ene benytter sig af modulsystemet. Såvidt programmerne yder lige godt, er dette succeskriterie opfyldt.

- **At Futharks usability øges gennem muligheden for at kalde modulers værdier gennem prik-notation.**

Hvorvidt dette successkriterie er opfyldt, vil blive bedømt gennem interviews med brugere, som arbejder med Futhark til dagligt.

1 Arbejdsopgaver

Projektet kan, udover udførelsen af en bachelorrapport, deles op i tre hoveddele:

Implementation af type-aliaser i Futhark, implementation af signaturer og moduler i Futhark, og slutteligt implementation af (SML)-funktorer i Futhark.

For hver af de tre delopgaver, gælder det, at

1. der skal udarbejdes et abstrakt syntaks-træ

2. opgaven skal implementeres i Futharks kodebase
3. løsningen skal testes og dokumenteres i projektrapporten.

1.1 Implementation af type-aliasser

Produkt: udvidelse af Futhark.

Ressourcekrav: Futhark-dokumentation. Denne er tilgængelig på [github.com\[Futhark\]](https://github.com/Futhark).

Interne afhængigheder: Ingen

Tidsforbrug: Uge 11 til 13.

1.2 Implementation af signaturer og strukturer

Produkt: udvidelse af Futhark.

Ressourcekrav: Futhark-dokumentation. Denne er tilgængelig på [github.com\[Futhark\]](https://github.com/Futhark). Desuden vejledning fra Futhark-designer Troels Henriksen.

Interne afhængigheder: Implementation af typealiasser.

Tidsforbrug: Uge 13 til 17

1.3 Implementation af funktorer

Produkt: udvidelse af Futhark.

Ressourcekrav: Futhark-dokumentation. Denne er tilgængelig på [github.com\[Futhark\]](https://github.com/Futhark). Desuden vejledning fra Futhark-designer Troels Henriksen.

Interne afhængigheder: Implementation af af signaturer og strukturer

Tidsforbrug: Uge 17 til 20

1.4 Tests

1.4.1 *Ad hoc* udvikling af simple testprogrammer, til løbende afprøvning modulsystem

Produkt: Testprogrammer skrevet i Futhark. Disse programmer vil have definerede inputs, og forventede outputs.

Ressourcekrav: Futhark-dokumentation. Denne er tilgængelig på [github.com\[Futhark\]](https://github.com/Futhark).

Interne afhængigheder: Testprogrammerne vil allesammen som udgangspunkt fejle, men flere og flere vil køre korrekt, jo mere færdig Futhark-moduler-implementationen bliver.

Tidsforbrug: Løbende under hele implementationsforløbet.

1.4.2 Udvikling af simple testprogrammer, til afprøvning af det færdige modulsytem

Produkt: Testprogrammer skrevet i Futhark.

Ressourcekrav: Futhark-dokumentation. Denne er tilgængelig på [github.com\[Futhark\]](https://github.com/Futhark).

Interne afhængigheder: Et færdigt modulsytem.

Tidsforbrug: Uge 19-20.

2 Overvejelser og tidsplan

Litteratur

References

Danny Gratzer, *A crash course on ML modules* <http://jozefg.bitbucket.org/posts/2015-01-08-modules.html>

HIPERFIT, APL section on Datalogisk Institut, KU. Official Futhark Github Repo and documentation <https://github.com/HIPERFIT/futhark>