

# **Opponering mot grupp 32**

## **“JSHC - a Javascript Haskell Compiler”**

*Grupp 15*

*Filip Reesalu, Jonas Bornold, Johan Gustafsson*

Vi har valt att dela upp opponeringen i tre delar. Vi börjar med att diskutera det generella intrycket följt av detaljkritik. Avslutningsvis presenterar vi helhetsintrycket i en sammanfattning.

## Allmänt intryck

Inledningsvis finner man innan att ha läst rapporten, som främmande för funktionella språk, att ämnet kan vara svårt att ta till sig. Det väl tilltagna kapitlet ämnat åt teori som tar en tredjedel av rapporten i anspråk lättar dock denna känsla. Här presenteras de två viktigaste aspekterna, språket Haskell och grunderna i hur en kompilator fungerar.

Rapportens struktur och upplägg är i helhet bra och varje avsnitts innehåll håller sig inom sina ramar. Sektionsdjupet upplevs ibland för djupt och varierar kraftigt över texten, vilket stundtals kan hämma läsarens fokus. Läsningen flyter däremot på bra då språket är lätt att ta till sig trots sin stundtals tekniska natur. Den gränsar dock ibland mot det informella med termer såsom “syntaktiskt socker”. I de flesta fall förklaras främmande facktermer för läsaren, men ibland kan man i texten finna dem introducerade för att först senare stöta på en förklaring.

Under rapportens läsning kan man undra hur arbetsprocessen gått till, och som läsare får man tyvärr inte sin nyfikenhet stillad. Rapporten sätter tidigt slutprodukten i fokus och med detta som den röda tråden hamnar vägen dit i skymundan och varken utvecklingsmiljö eller arbetsmodell nämns. Den istället insatte och resultatorienterade läsaren får desto mer att bita i då resten av texten i avsnitten för resultat och diskussion, de två sista tredjedelarna, är rapportens kärna. Dessa redogör för många implementationsdetaljer, jämförelser med befintliga kompilatorer, en gnutta kritik av Haskells grammatik såväl som förslag på förbättringar.

## Detaljer

**Abstraktet** är välskrivet och summerar rapporten med ett bra och tydligt språk, ett perfekt abstrakt vid första anblick. Dessvärre är allt innehåll i abstraktet inte en spegling av rapporten då exempelvis påståendet “The literature indicates that adding an extra layer of abstraction [...] can be a solution to this problem.” inte finns med i rapporten och därmed går inte litteraturhänvisningen att hitta. Ett resultat över jämförelsen som nämns i sista meningen hade även varit till fördel. Utöver det här saknas en summering på svenska.

I sektionen **Introduction** lyser problemformuleringen med sin frånvaro. Det går ändå att urskönja en problemformulering i introduktionen där den egentligen ordagrant är definierad. Ett tydligt problemformuleringsavsnitt är ändå avsaknat.

Projektets avgränsningar presenteras inte utan förblir odefinierade tills de i princip senare uppstår som en konsekvens av resultatet då delarna av Haskell som implementerats presenteras. Detta gör det svårt att se hur avgränsningar påverkat projektet i sin helhet. Däremot redogörs det dock för både informellt i texten och även formellt i bilaga vilka delar av språket som kompilatorn i

slutändan stöds.

Syftet är däremot mycket välskrivet och målet är klart definierat.

Vidare ger sektionen **Theoretical background** en god genomgång av Haskell-syntaxen, något som dock försvåras genom de formateringsfel som konsekvent löper genom rapporten för vissa tecken. En stor del av resultatet ägnas dock åt att beskriva exempelvis hur källkodstext tolkas och omformas till en abstrakt modell och därför kunde ett kortare stycke som introducerar reguljära uttryck och grunderna för lexikal granskning tjänat läsaren.

Fortsättningsvis beskriver sektionen **Method** vilken kringmjukvara som har använts för att underlätta arbetet. Beskrivningarna är väl skrivna men det känns som att den här metoden borde inkludera något mer om utvecklingsarbetet utöver hur arbetsfördelningen har varit.

För att beskriva slutprodukten radar **Result** upp funktionaliteten hos kompilatorn. Beskrivningen är bra upplagd med mindre introduktioner för varje del och detaljerade och djupgående beskrivningar. Det går inte att missa att texten inte är färdigskriven då fler än ett stycke endast har korta beskrivningar av vad som skall skrivas med öppningen "TODO". Som tidigare nämnt saknas en isolerad problemformulering, varför det också blir svårt att på något sätt återkoppla resultatet till ett ursprungsproblem. Kompilatorn var tänkt att erbjuda typsäker utveckling av webbapplikationer; Mycket vikt läggs vid hur kompilatorn fungerar tillsammans med Ymaccs-gränssnittet, men hur kompilatorn fristående skulle fungera praktiskt i en målmiljö skulle vara intressant att se. Något som även skulle vara uppskattat är en länk där man kan hämta kompilatorn för att titta på koden, utvärdera och testa den.

Avslutningsvis ger **Discussion** en bra översikt över de utvecklingsproblem som har uppkommit under utvecklingen samtidigt som önskad framtida funktionalitet både motiveras och beskrivs väl.

Utöver själva huvudrapporten tillkommer även två bilagor som i mer detalj ger en överblick av den syntaktiska kompabiliteten. Det här är en väldigt god idé och tabellerna är väl skrivna men tyvärr gör formateringsfelet listorna väldigt svårlästa.

## Sammanfattning

Från början av rapporten presenteras syftet klart och med slutprodukten - en Haskell kompilator i Javascript som i sin tur genererar Javascript - konstant i fokus genom rapporten förankras man trots att en klar problemdefinition uteblivit. Även fast metodavsnittet lämnar mer att önska presenteras ändå medlemmarnas uppdelning av arbetsbördan och de olika delmoment som identifierats. Den kringmjukvara som använts listas och refereras, så man kan fördjupa sig i komponenterna, och på så sätt får läsaren en klar bild över vilken mark kandidatarbetet från en början står på. Avslutningsvis kan man säga att det är en bra rapport som inte känns riktigt klar.