

Projektspecificering för DD143X
“A Prototype Implementation of Featherweight Java in Prolog”

Johan Öhlin
johanohl@kth.se
Handledare: Mads Dam, mfd@kth.se

12 februari 2010

Sammanfattning

Detta är projektspecifikationen för "A Prototype Implementation of Featherweight Java in Prolog" som är en del i kursen "Examensarbete inom datalogi, grundnivå", DD143X, på KTH vårterminen 2010. Specifikationen ska innehålla de viktigaste delarna för att projektet ska komma igång väl, och ha väldefinierade mål och en tidsplan. Målet är att påbörja en implementation av Featherweight Java, som är definierat av A. Igarashi, B. C. Pierce och P. Wadler i dokumentet "Featherweight Java: A Minimal Core Calculus for Java and GJ", i logikprogrammeringsspråket Prolog. På grund av den storlek som projektet ska ha kommer inte en full implementation göras. Utöver det ska Featherweight Java dokumenteras och vara en grunden till implementationen.

Innehåll

1 Bakgrund

En abstrakt specifikation av ett programmeringsspråk kan användas för att visa hur språket fungerar och bevisa att det fungerar så som man har tänkt sig. Featherweight Java är abstrakt beskrivning av kärnan, det mest fundamentala, av programmeringsspråket Java.

De abstrakta beskrivningarna är även bra för att kontrollera om en ny implementation eller ny version av språket kan utvecklas. Det kan göras genom att utöka specifikationen och bevisa att det man tänkt sig göra med språket fungerar.

Relationen mellan Featherweight Java och Java kan liknas vid vad λ -kalkyl är till Haskell[?], och har en semantik skriven som inferensregler på formen $\Gamma \vdash \Gamma_1 \wedge \Gamma_2 \wedge \dots \wedge \Gamma_\theta$. Dessa kan man möjligen göra en avbildning av på $\Gamma' \vdash \bigwedge_{\xi=1}^{\theta} \Gamma'_\xi$ och använda i andra språk.

2 Problemformulering

Målet är att undersöka om den abstrakta specifikation av ett programmeringsspråket Java, det vill säga Featherweight Java, kan användas för att skapa en implementation i logikprogrammeringsspråket Prolog.

2.1 Dokumentation

Featherweight Java finns beskrivet av A. Igarashi, B. C. Pierce och P. Wadler i deras rapport “Featherweight Java: A Minimal Core Calculus for Java and GJ”. Den rapporten ska i projektet dokumenteras och finnas som grund till andra delen av projektet.

Länk till rapporten i PDF från Pierces sida: www.cis.upenn.edu/~bcpierce/papers/fj-toplas.pdf

2.2 Implementation

Hur ska Featherweight Java implementeras i Prolog? I denna del av projektet ska en implementation byggd dokumentationen ske. Implementationen kan man dela in det i tre delar:

1. Semantikomskrivning
2. Parsing
3. Utökning och komplettering

Första delen kommer bestå att skriva om semantiken i Prolog. Rapporten innehåller regler som man enkelt kan skriva om till regler som används i Prolog. Steg två handlar om parsing, vilket handlar om att kunna köra kod skriven i Java direkt till Prolog. Den tredje punkten handlar om att komplettera implementationen med bland annat fler datatyper, och utöka det till något mer användningsbart och komplett.

Det är inte tänkt alla tre punkterna ska göras, utan främst första punkten på grund av tidsrestriktionen.

3 Projektplan

Första steget i projektet blir att förstå den rapport som A. Igarashi, B. C. Pierce och P. Wadler har skrivit. Detta görs genom att läsa den ett par gånger och eventuellt leta svar på frågor som kan uppstå. I första hand kommer svaren sökas med någon sökmotor på Internet, och, beroende på den frågan, sedan gå vidare till bibliotek eller handledare för mer hjälp.

Innan implementationsdelen påbörjas är det viktigt att jag förstått rapporten och har klart för mig hur jag ska påbörja min implementation. Sedan kommer diverse böcker om Prolog, Internet samt min handledare, som tidigare jobbat för SICS (som utvecklar SICStus Prolog[?]) samt har ett intresse för logikprogrammering[?], till hjälp när implementationsfasen väl är igång.

4 Tidsplan

Mellan vecka 4 och 18, ungefär tre och en halv månad, ska dokumentation, implementation och rapportskrivningen ske. Rapportskrivningen kommer ske under hela tiden, men de två sista veckorna innan rapporten ska lämnas är tid som bara spenderas till att komplettera och korrigera rapporten.

Dokumentationsfasen är den mindre delen, men ska inte gås igenom eftersom det är viktigt att förstå den delen innan implementationsfasen startas. Implementationsdelen är den större delen i hela projektet, och här finns mer som ska göras. Därför krävs antagligen implementationsdelen mer tid. Fem veckor för första delen och resterande sju för implementationen verkar vara en bra fördelning för att få tillräckligt med tid på båda..

Efter de första fem veckorna ska jag ha dokumenterat och ha en så pass god förståelse av Featherweight Java för att jag ska påbörja implementationen. Sedan har jag sju veckor på mig att göra ett så bra jobb som att uppfylla de krav som är beskrivet i del 2.2.

Rapporten ska lämnas in 3 maj, och examensarbetskonferensen kommer ske 10 maj.