

EXAMENSARBETE DrAST – An attribute debugger for JastAdd

STUDENT Joel Lindholm, Johan Thorsberg

HANDLEDARE Görel Hedin (LTH)

EXAMINATOR Ulf Asklund

Förstå din kompilator med DrAST

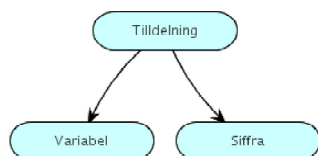
POPULÄRVETENSKAPLIG SAMMANFATTNING Joel Lindholm, Johan Thorsberg

Verktöget DrAST har skapats för att enklare förstå hur en kompilator faktiskt fungerar, den där magiska lådan som översätter kod till något datorn begriper. DrAST är en debugger, som illustrera hur kompilatorer ser ut på insidan.

En titt in i kompilatorn

Den kod som programmeraren skriver är faktiskt bara text och inget annat. Datorn förstår dock inte text, utan den måste översättas till maskinkod med hjälp av en kompilator. Kompilatorn måste först kontrollera att koden följer vissa regler. Vi kan jämföra detta med en grammatikgranskare av svenska språket. Om någon skriver "grisar hoppar sängar" ska kompilatorns analys förstå att språkreglerna inte följs.

En AST (Abstract Syntax Tree) är en trädstruktur som ofta används för att utföra dessa analyser. Låt oss kolla på ett exempelprogram. Koden "a = 2" betyder att "a" är en låda som innehåller en 2:a. En kompilator kommer att ta de viktigaste delarna i koden för att bygga en trädstruktur med noder (Figur 1). Tilldelning (likhetstecken) pekar på två saker: Variabel (lådan a) och Siffra (innehållet 2). Om en regel säger att Tilldelning ska peka på en Siffra och en Variabel, kan vi enkelt se om det stämmer i vårt träd.



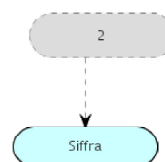
Figur 1: En AST som beskriver ett koden "a=2"

Problemet och lösningen

Till skillnad från Figur 1 så består ett program ofta av tusentals noder, och en överblick kan vara svårt att få utan hjälp. Det finns dock få hjälpmedel idag. Vårt nya verktyg DrAST är skapat för att fylla en del av detta tomrum för utvecklare av kompilatorer. DrAST kan nämligen "plocka ut" en AST från en kompilator och sedan illustrera den grafiskt för programmeraren.

Filtrering

DrAST har också andra egenskaper, än själva visualiseringen av en AST, som förenklar arbetet med kompilatorer. Som vi nämnt ovan kan ett program bestå av många noder. För att kunna fokusera på det intressanta i trädet kan det filtreras, med hjälp av regler som beskriver vilka noder som faktiskt ska visas. De noder som inte uppfyller alla reglerna gömmer DrAST helt enkelt. Figur 2 visar samma träd som Figur 1, men bara noder av typen Siffra visas.



Figur 2: Samma AST som i figur 1 men det är några noder som göms undan.

Vem kan ha nytta av DrAST?

Målet är att vårt verktyg ska kunna användas av många, allt från studenter som lär sig kompilator teknik till professionella som forskare och företag. Som sagt så hoppas vi att DrAST kan hjälpa till att fylla ett tomrum som funnits allt för länge.



Figur 3: En AST med över 60 000 noder som har filtrerats så den bara visar metoder för en specifik Java klass.