

Exjobbsspecifikation

Preliminär titel: Branching Tree Grammars in TREEBAG

Omfattning: 20p

Språk: engelska eller svenska

Mål: implementera "branching tree grammars" i systemet TREEBAG

Beskrivning:

Systemet TREEBAG använder sig av trädgrammatiker, bl.a. reguljära trädgrammatiker, för att generera objekt såsom strängar, bilder och träd. Det görs genom att tolka de genererade träden med hjälp av en algebra och visualisera resultatet på en såkallad display. Utöver trädgrammatiker, algebror och displayer finns det trädtransformationer, en fjärde typ av komponent som transformerar träd i andra träd. En viktig typ av trädtransformation som är implementerad i TREEBAG är klassen av "top-down tree transductions", nedan kallad *td-transformation*.

Det föreslagna projektet handlar om branching tree grammars, en klass av trädgrammatiker i vars deriveringar icketerminaler kan vara synchroniserade med varandra på ett antal nivåer, dvs. mer eller mindre starkt. Grammatiktypen har introducerats i artikeln [DE04] och är kraftfullare än de grammatiker som nuvarande TREEBAG ställer till förfogande. Det har dock visats i [DE04] att (och hur) en sådan grammatik kan översättas till befintliga komponenter, nämligen en reguljär trädgrammatik och en sekvens av td-transformationer. Antalet td-transformationer är lika med grammatikens *djup*, ett tal som bestämmer hur många synchroniseringsnivåer som finns i grammatiken. Projektets mål är att implementera branching tree grammars som en TREEBAG-komponent.

Principiellt finns det två tänkbara tillvägagångssätt:

1. Klassen av branching tree grammars implementeras från grunden enligt definitionen i nämnd artikel
2. Klassen implementeras genom att internt översätta en instans till en reguljär trädgrammatik och lämpliga td-transformationer. De där klasserna används sedan för att simulera grammatikens beteende

Sistnämnda metod har bl.a. fördelen att den intresserade användaren kan få möjlighet att inspektera de resulterande komponenterna för att lära sig något om sambanden mellan grammatiktyperna. Implementationen kommer att baseras på

den här idén om det inte under arbetets gång visar sig att en direkt implementation skulle innebära betydande fördelar. En branching tree grammar G av djup d översätts alltså till en reguljär trädgrammatik g och td-transformationer τ_1, \dots, τ_d så att $L(G) = \tau_d(\dots\tau_1(L(g))\dots)$. Som en ungefär mall på hur det i ett konkret fall kan se ut kan den befintliga implementationen av ET0L-trädgrammatiker i TREEBAG ses eftersom den beror på att översätta en ET0L-trädgrammatik till en (speciell typ av) reguljär trädgrammatik samt en td-transformation.

Istället för att direkt implementera den formella konstruktionen som beskrivs i [DE04] (med implementationen av ET0L-trädgrammatiker som förebild) bör arbetet dock ta hänsyn till effektivitetssynpunkter eftersom träden som produceras av td-transformationen τ_i ($1 \leq i < d$) enligt konstruktionen i [DE04] ofta är onödigt stora pga att de kan innehålla många grenar som τ_{i+1} inte beror på. Viktiga beståndsdelar av arbetet är således att

- göra sig förtrogen med branching tree grammars och deras översättning till reguljära trädgrammatiker och td-transformationer
- modifiera översättningen på så sätt att den förblir korrekt men kräver mindre resurser i de flesta praktiskt förekommande fall
- skapa en bra fungerande implementation (dvs. en implementation som både fungerar enligt den formella definitionen och är enkelt att använda)
- skriva en rapport som definierar och förklarar alla centrala begrepp, diskuterar de väsentliga idéerna bakom implementationen och presenterar exempel.

Referenser

- [DE04] Frank Drewes and Joost Engelfriet. Branching synchronization grammars with nested tables. *Journal of Computer and System Sciences*, 2004. To appear.