Direkt Profil

Ett verktyg för morfologisk analys av skriven inlärarfranska

Lisa Persson

Examensarbete för 20 p, Institutionen för datavetenskap,
Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 20 credit points
Department of Computer Science, Faculty of Science, Lund University

Direkt Profil
A tool for morphological analysis of written learner French
Abstract
In linguistic research on language acquisition, hypotheses have been put forward concerning how a learner's French spoken language is developed. Researchers seek now to evaluate these hypotheses with respect to written French. In this research a computerized tool would be a welcomed support. Automatizing the analysis would enable a larger material to be analysed. Such a tool could also be used for asserting the grammatical developmental level of written French. Based on previous research results, a decision tree has been developed. This is to be applied automatically on a larger amount of written learner texts.

This Master thesis is a development of a system for this automatization that has been started based on an existing decision tree for French verbs with pronominal subjects. The system, called Direkt Profil, is now fully implementing the decision tree and a prototype of the system is available via the web for linguists to test. The very core of the system is the so called parse engine, which is implemented in Java.

The report outlines how the parse engine works and how the decision tree has been adapted to the rule formalism in XML. A number of suggestions for further developments of the system is also given, since this Master thesis only is a part of the development of Direkt Profil. Some of these further developments have been started, but not finished. These are briefly described in the report.

Direkt Profil
Ett verktyg för morfologisk analys av skriven inlärarfranska
Sammanfattning
Inom den lingvistiska forskningen kring språkinlärning finns hypoteser för hur en inlärares franska talspråk utvecklas. Man vill nu försöka pröva dessa hypoteser för skriftspråk och behöver därför ett datoriserat verktyg som stöd i denna forskning. Genom att analysen automatiseras skulle ett större material kunna analyseras. Ett sådant verktyg skulle också kunna användas som nivåbedömning för att bestämma den grammatiska utvecklingsnivån i skrivna franska inlärartexter. Utifrån forskningsresultaten från talad franska har ett beslutsträd tagits fram. Detta ska nu appliceras automatiskt på en större mängd skrivna inlärartexter.

Detta examensarbete är en vidareutveckling av ett system för denna automatisering som påbörjats utifrån ett befintligt beslutsträd för franska verb med pronominella subjekt. Systemet, som kallas Direkt Profil, implementerar nu fullt ut nämnda beslutsträd och en prototyp av systemet finns tillgänglig via webben för lingvister att testa. Själva kärnan i systemet är den så kallade parsningsmotorn som är implementerad i Java.

Rapporten beskriver översiktligt hur parsningsmotorn arbetar samt hur beslutsträdet anpassats till en regelformalism i XML. Dessutom ges ett antal förslag på vidareutvecklingar av systemet, eftersom detta examensarbete endast är en del i utvecklingsarbetet med Direkt Profil. Några av vidareutvecklingarna har påbörjats, men inte slutförts. De beskrivs i korthet i rapporten.

Förord

Jag vill tacka Emil Persson för kreativt samarbete, Jonas Granfeldt och Suzanne Schlyter för givande diskussioner, min supporterklubb i Eker för uppmuntrande tillrop, Anso för moraliskt stöd och sist men absolut inte minst Pierre Nugues för eminent trespråkig handledning och inspiration.

"Honi soit qui mal y pense."

Lisa Persson, augusti 2004

.

Innehåll

1 Inledning

Att nivåbedöma uppsatser av elever i främmande språk är ett kvalificerat och tidsödande arbete. I alla tider har lärare varit tvungna att manuellt gå igenom varje enskild elevs uppsatser för att nivåbedöma dem. De mallar som finns till exempel hos Skolverket för nivåbedömning av uppsatser är främst kommunikativt inriktade. Nivåbedömning av en inlärares grammatiska färdigheter sker ofta utifrån ganska strikt uppbyggda uppgifter som inläraren får göra, men sällan utifrån fritt producerad text.

Forskningen om talspråksutveckling vid andraspråksinlärning visar att en inlärares språknivå visas genom vissa grammatiska indikatorer. Förekomst av få böjda verb eller endast enkla verbböjningar tyder obevekligen på en lägre språknivå än fler böjda verb och mer komplexa böjningsformer. Utifrån detta resonemang bör det gå att utforma enkla regler som hittar de strukturer som ligger till grund för bedömningen av vilken språklig nivå texten befinner sig på. (För att uppnå detta krävs dock mer lingvistisk forskning om hur det franska skriftspråket utvecklas hos inlärarna.)

I ljuset av detta vore det önskvärt med ett praktiskt användbart system för att utföra arbetet med att finna de strukturer som är relevanta för den språkliga nivån. Utvecklingen av ett sådant system har påbörjats inom ramen för ett forskningsprojekt i samarbete mellan Romanska institutionen vid Lunds universitet och Institutionen för datavetenskap vid LTH. Systemet går under namnet Direkt Profil och bestod i februari 2004 av ett 40-tal javaklasser, ett franskt lexikon i maskinläsbar form och en XML-kodad regelfil. En första (icke fullt exekverbar) prototyp av systemet hade utvecklats av Fabian Kostadinov och Jonas Thulin som ett projektarbete [17] i en kurs i datalingvistik vid Institutionen för datavetenskap. Det var inte helt stabilt att köra och de olika delarna var inte integrerade på ett tillfredsställande sätt.

Enligt 2004 års forskningsansökan till Vetenskapsrådet [10] lyder den övergripande målsättningen för det forskningsprojekt som detta examensarbete ingår som en del i:

> Projektet syftar till att utveckla en automatisk partiell morfosyntaktisk analys av skrivna inlärartexter på franska och utgör ett försök att kombinera dels empirisk språkinlärningsforskning och datorlingvistik, dels forskning och pedagogisk utveckling.
>
> Ett datorprogram, *Direkt Profil* håller på att utvecklas som skall användas som metodologiskt stöd för forskningen. *Direkt Profil* skall analysera ett 20-tal morfosyntaktiska fenomen (verbmorfologi, negationsplacering, objektspronomen, genuskongruens etc) i franska texter. Dessa element och strukturer har undersökts /[2]/ i talad franska hos vuxna svenska inlärare och de har visat sig utvecklas systematiskt över tid. De utgör därmed sk *utvecklingsgångar*.
>
> Målet med datorverktyget är dels att betydligt kunna öka den empiriska basen för forskningen om utvecklingsgångar för franska (i förhållande till talad franska), dels att kunna pröva hypoteser från forskningen om talspråk på skriftspråket.

Med Direkt Profil kan man på sikt göra en grammatisk nivåbedömning av fritt producerad text, något som traditionellt främst bedöms utifrån rent kommunikativa kvaliteter. Denna grammatiska bedömning är något mer avancerad än rent kvantitativa metoder som att "räkna antal fel" som ofta används vid grammatisk nivåbedömning. Genom att använda Direkt Profil får man en uppfattning om, inte bara vilken typ av fel som gjorts, utan även vilken typ av grammatiska strukturer inlärarskribenten använt sig av.

1.1 Bakgrund

Automatisk grammatikkontroll är inget nytt forskningsområde. Det finns ett antal olika projekt som i en eller annan form arbetar med det. I detta avsnitt ges en överblick över några sådana projekt.

Granska är ett grammatikkontrolleringsprogram för svenska som utvecklats vid KTH i Stockholm. Systemet använder både probabilistiska metoder och regler. Bearbetningen av texten sker i tre steg; tokenisering[1], PoS[2]-taggning och regelmatchning. Regelmatchningen består i sin tur i två steg; först hjälpregler som söker efter vissa grammatiska strukturer och sedan felregler (eng. *error rules*) som söker efter vissa specifika grammatiska problem [4]. Ett syfte med Granska är även att föreslå korrekta grammatiska konstruktioner som alternativ till de användaren skrivit. Granska riktar sig främst till personer som har svenska som modersmål [16].

CrossCheck är en utveckling av Granska och riktar sig till andraspråksinlärare. Målet är att det ska vara speciellt avpassat för skribenter som inte har svenska som modersmål och det utför både stavnings- och grammatikkontroll [8]. Man har där velat utveckla ett program som kan användas hela vägen i skribentens skapande, inte bara i granskningsprocessen. Syftet var att undvika ett program som ger så många falska alarm att själva skrivprocessen störtes och man ville undersöka förhållandet mellan andraspråksinlärares behov och Granskas möjligheter [16]. I studien fick användare av systemet ge sina synpunkter på dess användbarhet vad gäller skriftproduktion snarare än språkinlärning. Man hävdar att många tidigare studier fokuserat på talutvecklingen hos andraspråksinlärare och att skriftspråket därför kommit i skymundan. Granska ger olika precision- och recallvärden[3] för olika textgenrer. Anpassningen av Granska till andraspråksinlärare har gett vissa problem, exempelvis hög frekvens av falska alarm och låga recallvärden [16].

I Granska-projektet använder man sig av en PoS-taggare som visat sig effektiv [5] och man har även utvecklat Stomp (*Sto*ckholm *M*atching *P*art-of-speech Tagger), en PoS-taggare som uppvisat goda resultat för okända och sammansatta ord [21].

Begreppet *grammar checking* kan definieras lite olika, men ett sätt är att "kontrollera en text och leta efter fel som en grammatikbok skulle diskutera" [12]. Microsoft Research har dragit sitt strå till stacken med de grammatikkontroller som finns i Word. Microsofts NLP System är en komplett parsare för naturligt språk. Syftet med Microsoft NLP är att naturligt språk ska kunna både analyseras och genereras, från och till en betydelserepresentation.

*Word 97*s grammatikkontroll syftar till att föreslå användaren alternativa lösningar. Man använder begreppet *relaxed parsing* för meningsfull parsning av inkorrekt eller dåligt formulerad inputtext som exempelvis saknar kongruens vad gäller numerus eller person. I sådana fall används något man kallar *fitted parsing* vilket innebär att man producerar en approximativ parsning som används som input till de återstående stegen i processen [14]. Word 97s grammatikkontroll riktar sig till användare som har språket som modersmål och man har alltså inte tagit särskilda hänsyn som krävs vid andraspråksinlärning.

FreeText är ett projekt som utvecklats vid LATL[4] vid Genèves universitet. Det riktar sig direkt till andraspråksinlärare av franska. Den syntaktiska analysen genomförs av Fips, en redan existerade parsare som anpassades för andraspråksinlärare [22]. Programmet är utformat som uppgiftsbaserade aktiviteter som användaren utför och sedan får feedback på. FreeText innehåller dock mycket mer än bara diagnos, exempelvis en talsynteserare (*speech synthesiser*) och en

[1] Se ordlistan på sidan 58.
[2] Se ordlistan på sidan 57.
[3] Se ordlistan på sidan 57
[4] Laboratoire d'Analyse et de Technologie du Langage

meningsomformulerare. Målet sägs vara ett komplett NLP[5]-baserat CALL[6]-system för franska [9].

Rapid Profile är ett forskningsprojekt vid Paderborns universitet som liknar Direkt Profil i så måtto att det inte syftar till att korrigera inlärarnas språk utan endast att nivåbedöma det. Det riktar sig dock inte till franska utan till tyska och engelska och inte till skriftspråk, utan talspråk (Pienemann/Mackey 1992, refererad av Granfeldt 2004 [10]). Det är Rapid Profile som varit den lingvistiska inspirationskällan till Direkt Profil.

Fastus är ett amerikanskt projekt vars huvudintresse är att extrahera information ur naturligt språk för att lagra i en databas. Man använder sig av mönstermatchning snarare än fullständig parsning och det har visat sig fungera väl för att hitta svar på frågor om vem, var och när [13].

Ur framställningen ovan kan urskiljas framförallt tre skillnader mellan Direkt Profil och många av de befintliga programmen och forskningsprojekten:

1. Direkt Profil syftar till att detektera utvecklingsrelevanta grammatiska strukturer i inputtexten och hittar därmed såväl korrekta som felaktiga struktuer. Många av de andra programmen vill hjälpa skribenten till en bättre text, det vill säga hitta fel och föreslå förbättringar, till exempel Granska och Word 97.

2. Direkt Profil är framtaget med andraspråkstexter direkt i åtanke. Andra program riktar sig till texter där författaren har språket som modersmål och alltså skrivit nästan korrekt grammatik, till exempel Granska, Word 97 och Fips. Åter andra har sökt anpassa sin grammatikkontroll till andraspråksskribenter genom så kallade *relaxed constraints* (se nedan i avsnitt 1.3 på sidan 13), exempelvis CrossCheck och FreeText.

3. Direkt Profil använder partiell parsning, det vill säga annoterar endast de strukturer som vi specifikt vill studera. Det gör även Fastus och Granska, men i det sistnämnda projektet används en PoS-taggare, det gör vi inte i Direkt Profil. Många andra program genomför en komplett parsning, exempelvis FreeText och Microsoft NLP.

Nivåbestämningsreglerna som används i Direkt Profil är baserade på Bartning/Schlyters arbete om franska utvecklingsstadier [2]. Enligt dessa hypoteser använder språkinlärare på olika nivåer olika syntaktiska mönster. Hypotesen är att dessa mönster bildar utvecklingsgångar[7], det vill säga att de förändras på ett givet sätt över tid och i takt med att språkinläraren förbättrar sina kunskaper i språket.

1.2 Syfte

Direkt Profil har till uppgift att dels automatiskt analysera ett antal lingvistiska fenomen som är relevanta för forskningen om skriftspråkutvecklingen i franska, dels fungera som ett praktiskt redskap för svenska franskinlärare. Systemet kan alltså sägas ha två långsiktiga användningsområden:

1. Det ena långsiktiga målet med Direkt Profil är att det skulle kunna användas till språktester. Ett sådant program skulle kunna hjälpa språkinlärare med svenska som modersmål att bestämma vilken språknivå de befinner sig på utifrån texter på franska som de själva producerat.

[5] Natural Language Processing.
[6] Computer Aided Language Learning.
[7] Se ordlistan på sidan 58

2. Det andra långsiktiga målet med Direkt Profil är mer teoretiskt orienterat; att systemet ska kunna användas i forskningssyfte. Forskningsgruppen har själv formulerat nyttan med ett program som inte endast är inriktat på talad franska [11]:

   > I franska är talspråket strikt skilt från skriftspråket. Vissa markeringar som ingår i utvecklingsgångar, tex förfluten tid, genus, pluralartikel, objektspronominas plats etc är "hörbara" och vi förväntar oss ingen egentlig avvikelse här från utvecklingsgångarna i tal eftersom det finns en överensstämmelse mellan tal och skrift. Andra markeringar finns bara i skrift (dvs bara som grafem). I denna kategori ingår de grammatiska ändelserna (*-s, -e, -es, -ent*) och deras utveckling kan alltså bara undersökas i skriven franska.

   Lingvisterna skulle genom att använda Direkt Profil som verktyg i sin forskning slippa annotera[8] och räkna strukturer manuellt. Detta är huvudsyftet med systemet. Det vore önskvärt om lingvisterna själva på lång sikt kunde laborera med inställningarna i Direkt Profil.

Det kan särskilt understrykas att det är punkt nummer två ovan som Direkt Profil främst utvecklas för. Från lingvistiskt håll vill man rent konkret se ett system som kunde ta en text som input och producera annoterad text med information om antalet detekterade strukturer i varje kategori som output. Nedan följer två autentiska exempel på texter som inlärare i franska skrivit. Markeringarna i texterna visar på vissa grammatiska strukturer som är fokus för intresset i Direkt Profil. I bilagorna T och U finns samma texter med markeringarna i färg.

Den första är skriven av en inlärare vi kallar Daniel och som befinner sig på språknivå 2. I Daniels text finns tre grammatiska fenomen uppmärkta. Med lila färg visas lexikala verb med kongruens[9] (6 förekomster), med grön färg *être/avoir* med kongruens med sitt pronomen (4 förekomster) och med röd färg icke-finita[10] konstruktioner (4 förekomster):

Je m'appelle Daniel et j'ai 18 ans. Je suis en terminal à la école. Je commencer lire francais dans six grade. Ma mère as habit en France. Nous voyager à Paris deux. J'aime parle francais. Dans l'école je ne apprendre rien francais. Sur quelque ans je voyager au francais Martinique et apprendre parle francais. À maison je parfois regarde un film francais. Je ne ai pas de francais copains. Pour a lire francais trois ans je ne pas beaucoup parle francais. J'aime parle francais, beaucoup. Je crois un voyager à Paris c'est tres bon pour étudie francais.

Den andra texten är skriven av en inlärare som vi kallar Rita och som placerats på språknivå 4. I Ritas text finns också tre grammatiska fenomen utsatta. Med gul färg visas *passé composé*-konstruktioner med kongruens (7 förekomster), med ljusgrön färg visas *imparfait*-konstruktioner (1 förekomst) och med grå färg modalt hjälpverb med kongruens och efterföljande infinitiv (1 förekomst):

J'ai commencé à étudier le francais à l'âge de treize ans. Je me suis toujours intéressée aux langues, et j'ai trouvé le francais une langue très belle. Au début la beauté de la langue

[8] Se ordlistan på sidan A.
[9] Se ordlistan på sidan 56
[10] Se ordlistan på sidan 56

était la raison de mon intérèt. Il n'a pas pris longtemps avant que j'ai trouvé que des études d'une langue étrange m'a donné beaucoup plus que beauté. En lisant des textes et livres en francais, je me suis sentie comme si j'étais dans un monde completement différent. En parlant j'ai découvert un sentiment de comment il doit être pour un enfant qui apprend sa langue maternelle.

Skälet till att Daniels text bedöms som nivå 2 och Ritas text som nivå 4 är att de olika grammatiska fenomenen som markerades ovan sammantaget bildar en helhetsbild av textens nivå. Detta har systematiserats av Schlyter [20] i en tabell ur vilken delar redogörs för nedan:

Språknivå	1	2	3	4	5	6
Inlärartid, icke lärarledd	1-5 mån	4-9 mån	8-13 mån	12-24 mån	3 år	Mer än 3 år
Studietid	Nybörjare	1-2 skolår	3-5 skolår	1-2 universitetsterminer	3-4 universitetsterminer	Fransklärare
Lärarledd studietid	25-100 timmar	80-200 timmar	150-500 timmar	300-800 timmar	—	—
Andel satser med verb	20-40%	30-40%	50%	60%	70%	75%
Andel finita verb	50-75%	70-80%	80-90%	90-98%	100%	—
*Kopulaverb (*être*) och andra hjälpverb (*avoir/aller*)*	Formler utan opPoSitionsförståelse: *j'ai/c'est*	Förståelse för opPoSitioner: *j'ai - il a*	Enstaka misstag: *je va, je a*	—	—	—

Daniel har relativt många icke-finita verb. Ju fler icke-finita verb, desto lägre nivå, vilket framgår i tabellen. Det finns heller inga förekomster av komplexare verbformer som uttrycker andra tempus än presens. Rita har inga icke-finita verb i sin text. Hon har dessutom en förekomst av imparfait, vilket tyder på hög språknivå, eftersom det är en komplexare verbform. Daniel använder inte heller några *passé composé*-former, och definitivt inte några andra konstruktioner för att uttrycka förfluten tid. På det här sättet kan en uppmärkning (annotering) av inlärartexter användas i kombination med tabellen för att nivåbestämma dem. Syftet på lång sikt är att detta ska kunna göras automatiskt med Direkt Profil.

Konkret skulle detta ske genom igenkänning och PoS-annotering av de aktuella strukturerna. I bilagorna B till och med I visas den önskade outputen från Direkt Profil. Målet är att systemet från inputtexter som Daniels och Ritas ska kunna producera sådana annoteringsfiler. Det är dessa filer som ligger till grund för den mer läsvänliga färgkodade outputen i de två exempeltexterna ovan.

Det långsiktiga datalogiska syftet med projektet beskrivs i forskningsansökan till Vetenskapsrådet [10]:

> För den datorlingvistiska delen är målet att kunna representera kunskapen från ut-

vecklingsgångarna i form av en lokal grammatik, tillräckligt smidig för att analysera uttryck från nybörjare till avancerad inlärare. Programmet skall kunna detektera former i inlärartexter och annotera dem med en typ som motsvarar något av kriterierna från utvecklingsgångarna.

Syftet med detta examensarbete kan sammanfattas med att den befintliga versionen av Direkt Profil utvecklas och förbättras. Detta görs genom att de nivåbestämningsregler för verb som finns framtaget inom ramen för forskningsprojektet helt och hållet implementeras. Det konkreta målet är att systemet ska kunna identifiera, klassificera och räkna mönster automatiskt och utifrån den resulterande utvärderingen kunna nivåbestämma den enskilda inläraren, samt att systemet ska vara så stabilt att det går att använda i den lingvistiska forskningen kring franska utvecklingsgångar.

1.3 Metod

I detta avsnitt beskrivs först Direkt Profils språkbehandlingsmässiga metod att arbeta och sedan den metod som vi utvecklare av systemet använt. Det finns huvudsakligen två metoder för grammatisk parsning:

1. *Fullständig parsning.* Med parsa och parsning brukar normalt avses att bygga hela satsdelsträd eller på annat sätt helt och hållet analysera den grammatiska strukturen i den aktuella satsen som parsas. Denna metod används, som tidigare nämnts, av Micorsoft NLP och FreeText.

2. *Partiell parsning.* Man behandlar endast vissa språkliga fenomen och hoppar över sådant som är "ointressant". Regelmatchning är ett sätt att göra detta. Partiell parsning används bland annat i Granska och Fastus.

Vi har valt att använda den sistnämnda av dessa två metoder. Skälet härtill är främst att fullständig parsning är generellt svårt. System som utgår från en parsare anpassad efter skribenter med det aktuella språket som modersmål använder sig ofta av *relaxed constraints*, när de ska anpassa sin parsare till inlärare med ett annat modersmål. Det innebär att man utgår från en parsare som kan parsa de grammatiska strukturerna i korrekta, eller nästan korrekta, meningar. Denna är dock inte mycket behjälplig när det gäller att parsa fragmentariska meningar, meningar som saknar verb, eller liknande. Därför ersätter man de regler man tidigare haft i sin grammatik, med *relaxed constraints*, så att parsaren accepterar allt fler meningar som endast delvis är korrekta. Man "lättar på kraven" helt enkelt. Direkt Profil däremot är ett system som direkt riktar sig till skribenter med ett annat modersmål än det aktuella språket.

En fördel med vårt system är att de mönster Direkt Profil skall leta efter går att beskriva i en lokal grammatik. Med lokal grammatik avses här grammatiska regler som inte är heltäckande, utan endast behandlar en begränsad mängd av de fenomen som finns i språket. Direkt Profil skall alltså inte parsa hela meningar utan endast skanna texten och leta efter vissa lokala strukturer (exempelvis delar av en verbfras). Därigenom skulle systemet även kunna klara av att parsa fragment som inte är korrekt byggda.

Det språkbehandlingsmässiga tillvägagångssättet är att annotera specifika grammatiska strukturer i en text. Annoteringen av de funna strukturerna i texten sker med hjälp av PoS-taggar (Part-of-Speech-taggar, se ordlistan på sidan 57) och detektering av verbgrupper. Direkt Profil annoterar de segment som ligger till grund för bedömningen av inlärarens språkprofil. PoS-taggningen och strukturdetekteringen är sammansmält i ett steg i Direkt Profil till skillnad från många andra projekt, exempelvis Granska, där de två stegen är åtskilda och konsekutiva. Att vi

valde denna metod berodde främst på att vi inte hade tillgång till någon tillfredsställande PoS-taggare. Det har heller inte funnits planer på att utveckla någon inom projektet eftersom det inte tycks behövas; PoS-taggare kan också ge dåliga resultat för inkorrekt input, vilket det ju ofta är fråga om i Direkt Profil. Detektion av satser och fraser som motsvarar de olika inlärningsnivåerna sker genom en enkel PoS-taggningsstrategi parallellt med frasigenkänning (mönstermatchning). Detta är möjligt tack vare att beslutsträdet endast behandlar en mycket lokal kontext åt gången, genom vilket tvetydigheter i PoS-taggningen i hög grad kan undvikas.

Metoden som Direkt Profil använder för att hitta de aktuella grammatiska strukturerna är att det implementerar ett beslutsträd som tagits fram utifrån forskningsresultaten om utvecklingsgångar för talad franska [2]. Det har fått sitt nuvarande utseende genom en process där många personer i forskningsgruppen deltagit, främst lingvisterna Jonas Granfeldt och Suzanne Schlyter, men även datalogerna Fabian Kostadinov, Jonas Thulin, jag själv och systemvetaren Emil Persson. Utvecklingen av trädet är av allt att döma inte färdig ännu. Det kan komma att modifieras i framtida versioner av Direkt Profil. Beslutsträdets utseende och användning beskrivs närmare i nästa kapitel.

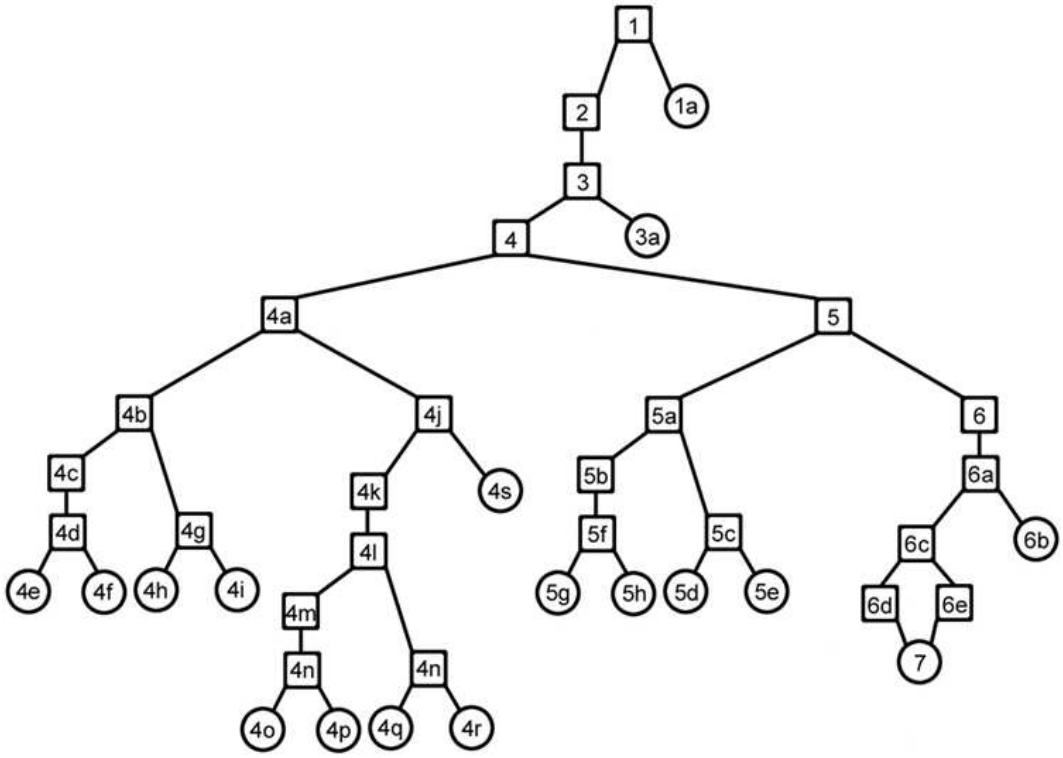
Utvecklingen av Direkt Profil har skett i samarbete med Emil Persson vid Stockholms universitet. Han har huvudsakligen stått för programmerandet av parsningsmotorn, särskilt i den nya versionen av systemet och utvecklandet av webbgränssnittet, medan jag själv har arbetat med färdigställandet av den äldre versionen, anpassning av det befintliga beslutsträdet till regelformalismen, utvecklandet av regelfilerna, testning och resultatberäkningar samt konverteringen av lexikonet. Vi har hjälpts åt med algoritmförbättringar och felsökning samt den övergripande utformningen av systemet. Därför kommer jag i fortsättningen att skriva “vi”, snarare än försöka särskilja exakt vad som gjorts av vem, en särskiljning som för övrigt knappast låter sig göras.

Programmeringsspråket vi använt är Java. Valet var naturligt av flera skäl; det befintliga systemet som vi utgick ifrån var implementerat i Java, det är ett språk som vi båda behärskar sedan tidigare och det är plattformsoberoende och lämpar sig därför väl för ett system vars framtida användning vi inte kan se vidden av. Koden har skrivits i vanliga texteditorer, främst *emacs*, och delats via en CVS[11]-server.

[11] Concurrent Versioning System

2 Beslutsträdet

Det beslutsträd som ligger till grund för hur Direkt Profil arbetar har utarbetats av professor Suzanne Schlyter och fil. dr. Jonas Granfeldt vid Romanska institutionen vid Lunds universitet[12]. Trädet är den kunskapsrepresentation som används för att parsa satserna i texten och beskriver hur det avgörs vilken typ av struktur en viss konstruktion utgör. I trädet finns ett antal räknare angivna. Dessa räknas upp när deras motsvarande strukturer påträffats i texten. Än så länge är beslutsträdet mycket rudimentärt till sin utformning och kommer att modifieras i framtida versioner av Direkt Profil. Dessutom är fler träd planerade att implementeras, som behandlar exempelvis negationer, klitiska pronomen[13] eller kongruenskontroll mellan adjektiv och substantiv.



Figur 1: Trädet sådant det slutgiltigt såg ut från Romanska institutionen

1. Är ordet ett subjektspronomen?
1a. Gå till nästa ord.
2. Öppna ett fönster om fem ord.
3. Finns det ett verb i fönstret?
3a. Räkna upp en förekomst av fras utan verb.

[12]Trädet har inte publicerats. Det är framtaget på grundval av Bartning/Schlyters arbete om utvecklingsgångar [2]
[13]Se ordlistan på sidan 56

4. Är verbet en form av *être/avoir*?
4a. Är verbet böjt i presens?
4b. Finns det ett verb i particip inom tre ord framåt?
4c. Räkna upp en förekomst av *passé composé*[14].
4d. Råder kongruens mellan pronomenet och *être/avoir*-verbet?
4e. Räkna upp en förekomst av *passé composé* med kongruens[15].
4f. Räkna upp en förekomst av *passé composé* utan kongruens.
4g. Råder kongruens mellan pronomenet och *être/avoir*-verbet?
4h. Räkna upp en förekomst av *être/avoir* i presens med kongruens.
4i. Räkna upp en förekomst av *être/avoir* i presens utan kongruens.
4j. Är verbet böjt i *imparfait*[16]?
4k. Räkna upp en förekomst av *imparfait*.
4l. Finns det ett verb i particip inom tre ord framåt?
4m. Räkna upp en förekomst av *plus-que-parfait*[17].
4n. Råder kongruens mellan pronomenet och *être/avoir*-verbet?
4o. Räkna upp en förekomst av *plus-que-parfait* med kongruens.
4p. Räkna upp en förekomst av *plus-que-parfait* utan kongruens.
4q. Räkna upp en förekomst av *être/avoir* i *imparfait* med kongruens.
4r. Räkna upp en förekomst av *être/avoir* i *imparfait* utan kongruens.
4s. Räkna upp en förekomst av *être/avoir* i andra tempus/modus.
5. Är verbet ett modalt hjälpverb?
5a. Finns det ett verb i infinitiv inom tre ord framåt?
5b. Räkna upp en förekomst av modalt hjälpverb + verb i infinitiv.
5c. Råder kongruens mellan pronomenet och hjälpverbet?
5d. Räkna upp en förekomst av modalt hjälpverb med kongruens.
5e. Räkna upp en förekomst av modalt hjälpverb utan kongruens.
5f. Råder kongruens mellan pronomenet och hjälpverbet?
5g. Räkna upp en förekomst av modalt hjälpverb med kongruens.
5h. Räkna upp en förekomst av modalt hjälpverb utan kongruens.
6. Konstatera att det rör sig om ett lexikalt verb.
6a. Är verbet personböjt?
6b. Räkna upp en förekomst av icke-personböjt verb.
6c. Råder kongruens mellan pronomenet och verbet?
6d. Räkna upp en förekomst av lexikalt verb med kongruens.
6e. Räkna upp en förekomst av lexikalt verb utan kongruens.
7. Summera räknarna 6d och 6e för att få den totala summan av personböjda verb.

Olika tempus kontrolleras endast för *être/avoir* i trädet. Det är presens och *imparfait* som särskljs från övriga tempus/modus. Detta är avsiktligt eftersom forskningen visar att det främst är *être/avoir* som hos inlärare förekommer i *imparfait* [2].

[14] Se ordlistan på sidan 57
[15] Se ordlistan på sidan 56
[16] Se ordlistan på sidan 56
[17] Se ordlistan på sidan 57

2.1 En guidad tur genom trädet

Vid första anblick kan trädet te sig komplicerat, varför en guidad tur genom dess olika grenar kan vara behjälplig för läsare som inte är lingvistiskt bevandrade. Noderna i trädet motsvarar ja/nej-frågor och att gå till vänster nedåt i trädet är detsamma som att svara ja på en fråga och vice versa.

Antag att vi med hjälp av trädet ska parsa den första meningen från Daniels text: *Je m'appelle Daniel et j'ai 18 ans*. Den första noden innebär att man kontrollerar om det aktuella ordet är ett subjektspronomen. *Je* är ett subjektspronomen och därför går man till vänster i trädet och söker efter ett verb inom ytterligare fem ord framåt i meningen. Det första verb som hittas är *appelle* och det finns inom fem ord från pronomenet. Därför svarar man ja på frågan i nod 3 och går till vänster nedåt i trädet. Nod 4 i frågar om verbet är en form av *être* eller *avoir* och därför blir svaret nej. Man går vidare ner till höger och i nod 5 ställs frågan om det är ett modalt hjälpverb som påträffats. De modala hjälpverben är sju till antalet och består av verben *vouloir, pouvoir, savoir, devoir, faire, laisser* och *falloir*. Svaret på frågan är nej så man går till höger i trädet till nod 6 där man kan konstatera att det rör sig om ett lexikalt verb. Nod 6a frågar nu om verbet är personböjt. Verbet *appelle* är personböjt efter första person singularis, så svaret på frågan är ja. Nod 6c frågar slutligen om det råder kongruens[18] mellan verbet och dess pronomen. Eftersom *je* är ett pronomen i första person singularis går vi till vänster och räknar upp räknaren som finns i det lövet.

Man har nu nått ett löv i trädet och arbetsgången börjar därför om från nod 1 igen. Nästa ord att parsas är *Daniel*, men det är inget pronomen. Därför hamnar man i löv 1a varifrån man endast går vidare till nästa ord. Detta upprepas även för *et* och det är först när man kommer till *j* som man går till vänster och hamnar i nod 2. På samma sätt som tidigare öppnas nu ett fönster om fem ord och ett verb påträffas ju redan vid nästa ord *ai*. Svaret på frågan i nod 3 är alltså ja och man fortsätter vidare åt vänster till nod 4. Även här svarar vi ja eftersom *ai* är en böjd form av verbet *avoir*. Nod 4a frågar nu om verbet man hittat är böjt i presens och svaret på den frågan är ja, så man hamnar i nod 4b. Här ställs frågan om det finns ett verb i particip inom tre ord framåt. Det gör det inte och därför går man vidare ner till höger. Nod 4g frågar nu om det råder kongruens mellan pronomenet och verbet. Svaret är ja, man går till vänster och hamnar i lövet 4h i trädet och räknar upp en annan räknare, nämligen en förekomst av korrekt böjd *être/avoir*. Alla löv i trädet inkrementerar räknare, men räknare räknas även upp på andra ställen än i löv. Meningen har nu parsats på de sätt som visades i avsnitt 1.2 på sidan 11: *Je m'appelle* *Daniel et* *j'ai* *18 ans.*

2.2 Problemet med bakåtreferenser

Den första versionen av Direkt Profil är en direkt påbyggnad av det arbete som Kostadinov/Thulin påbörjade inom ramen för forskningsprojektet under hösten 2003 [17]. Den har genomgått en hel del testning och rättande av fel under våren 2004, varför den är att betrakta som relativt robust som prototyp. Skälet till att vi valde att fortsätta med en helt ny avknoppning är att den första versionen lider av vissa brister som cementerats redan i inledningsskedet av Kostadinovs/Thulins arbete och som endast svårligen skulle låta sig ändras nu när arbetet var så långt framskridet. Den allvarligaste av dessa brister var att möjligheter till bakåtreferenser saknades. Kostadinov/Thulins regelformalism tillåter nämligen inte att variabler sparas undan eller att man backar och går uppåt i beslutsträdet. För att kunna kontrollera kongruens mellan ett personböjt verb och dess subjektspronomen, var man tvungen att veta "var man kom ifrån" i trädet. Den

[18] Se ordlistan på sidan 56.

enda möjligheten att veta det var att varje enskild nod var unik, inte bara i fråga om vilken regel den innehåller, utan även i fråga om vilket pronomen som orsakat att man hamnat just där.

Ett exempel är när man ska parsa *j'ai 18 ans* (som beskrevs ovan). När man nått till nod 4g ska man kontrollera om verbet är korrekt böjt efter pronomenet *je*. Härifrån finns dock ingen information att tillgå att det är just *je* man ska kontrollera kongruens mot. För att kunna veta vilket pronomen man ska kontrollera kongruens mot lades extraregler till efter första noden som kontrollerade om pronomenet var första person singularis. I såfall gick man vidare till regeln "je_verb1", i annat fall vidare till nästa extraregel som kontrollerade om pronomenet var andra person singularis. I så fall gick man vidare till regeln "tu_verb1" och så vidare. Det som kontrollerades i merparten av de efterföljande reglerna var exakt samma sak för de olika pronomina och hade alltså inget med kongruens att göra. Det var först när man kom till en kongruenskontrollerande nod som det blev intressant att veta om man var i exempelvis "je_accord_etre_avoir" eller "tu_accord_etre_avoir". Denna lösning ger alltså en multiplicering av trädet med åtta redan efter den andra noden (ett träd för varje pronomen) och är ohållbart i längden. På något sätt var vi alltså tvungna att spara information från tidigare noder. Denna information lagras i annoteringen som beskrivs i nästa kapitel. Det påverkar även reglerna på det sättet att man måste kunna ange om en parsningsregel ska kontrollera kongruens och i så fall utifrån vilka särdrag[19] (beskrivet nedan i avsnitt 4.6 på sidan 28).

[19] Se ordlistan på sidan 58

3 Annoteringsformatet

För att kunna lösa problemet med bakåtreferenser måste vi på något sätt spara undan information om tidigare matchningar längre upp i trädet. Detta för att man ska kunna gå tillbaka till tidigare matchningar, när man vill kontrollera kongruens. De ord man tidigare parsat behöver alltså på ett eller annat sätt annoteras.

Swedish Treebank är ett svenskt projekt som annoterar[20] en stor korpus[21] på svenska. Det är ett samarbete mellan Växjö universitet, KTH och Stockholms universitet. En av de frågor som diskuterats inom detta projekt är vilket annoteringsformat som bäst lämpar sig för en svensk korpus. Fyra olika format jämfördes, nämligen MAMBA, SWECG, SynTag och S-CLE. De två förstnämnda visade sig vara mest lämpade för ändamålet [18].

Det finns även andra annoteringsmetoder och vi har valt att implementera en förenklad form av det format som utvecklats inom projektet Granska vid KTH [3]. I sitt förslag beskriver man en uppdelning av informationen i tre olika filer, nämligen tokenfil, taggningsfil och frasinformationsfil. Vi använder de tre filtyperna i så gott som omodifierad form. De är kodade i XML-format liksom de andra språkspecifika filerna i Direkt Profil och vi kallar dem för *text.xml*, *pos.xml* respektive *span.xml*. Dessa filer utgör i princip den output som önskas av systemet, givet en inmatad text.

Ett token är en enskild enhet i en ström av tecken. Här utgörs tokens typiskt av ord, men kan även vara kommateringstecken till exempel punkter. Detta skiljer sig alltså från den definition av token (där ett token utgörs av alla på varandra följande icke-blanka tecken) som dataprogrammeraren är van vid. Apostrofer är också egna tokens, så meningen *Je m'appelle.* innehåller fem tokens; de tre orden, apostrofen och punkten[22].

3.1 Tokenfilen

Tokenfilen (*text.xml*) ger varje token ett unikt id. Taggarna i den filen heter *token* och har endast ett attribut, nämligen *id*. Värdet på attributet *id* används av annoteringstaggarna i de andra filerna för att referera till ett visst token. En *token*-tagg som ger ett unikt id till tokenet *je* ser typiskt ut så:

```
<token id="token.0">je</token>
```

Vår definition av tokens har varit ändamålsenlig och inte gett några problem i de tester vi gjort. Vi är dock medvetna om att vissa ord i franskan innehåller tecken som fungerar som tokenavgränsare. Ett exempel är ordet *aujourd'hui* som kommer att delas upp i tre tokens och därmed inte kunna hittas i lexikonet. Detta kan i framtiden lösas genom en analog implementation av den lösning som föreslås för chunks (se avsnitt 8.1 på sidan 45). Exempelutdrag ur tokenfiler återfinns i bilagorna B och F.

3.2 Taggningsfilen

Taggningsfilen (*pos.xml*) innehåller information om de olika tokenas PoS-annotering. Filens taggar heter *tag* och har fyra attribut. Attributet *id* ger denna annoteringstagg dess unika id som refereras till av de andra annoteringsfilerna. Attributet *tok_id* refererar till ett av de id som finns i tokenfilen. Attributen *lemma* och *pos_name* ger grammatisk information om tokenet; *lemma*s

[20] Se ordlistan på sidan 56
[21] Se ordlistan på sidan 56
[22] Se ordlistan på sidan 58

värde är tokenets lemma och *pos_name*s värde är en sträng ur vilken åtminstone ordklass kan utläsas. Strängen är uppbyggd på liknande sätt som den grammatiska informationen i lexikonet. Alla koderna är separerade med punkter (istället för kolon och plustecken, se avsnitt 5.1 på sidan 31) och den första koden står för ordklass och de efterföljande för numerus, person, genus och så vidare. En *tag*-tagg som innehåller lemma- och PoS-information för tokenet i förra exemplet ser ut så:

```
<tag id="tag.0" tok_id="token.0" lemma="je" PoS_name="Pro.SG.P1"/>
```

KTHs annoteringsförslag [3] anger två olika former för hur man hanterar att flera tokens har samma PoS-annotering, nämligen:

1. Alla tokens får en unik tagg med samma id som tokenet har. I taggningsfilen är alltså taggarna listade utifrån sina tokens.
2. De tokens som har samma PoS-annotering och därmed samma tagg, listas tillsammans under denna. I taggningsfilen är alltså tokena listade utifrån sina taggar.

Vi har valt att följa det första av dessa alternativ. Om det finns flera möjliga PoS-tolkningar i lexikonet väljs den som bäst matchar regeln[23]. Vår taggningsfil skapas nämligen inte förrän vid regeltillämpningen, vilket också resulterar i att de tokens som inte uppfyller någon regel inte får några värden på sina *lemma-* och *PoS_name*-attribut. Vi har diskuterat om detta kan komma att bli en begränsning för systemet, men kommit fram till att så inte är fallet. Om alla tokens skulle ges PoS-information skulle vi lagra en stor mängd redundant information som aldrig används i systemet. Direkt Profil inte är något fullständigt parsningsverktyg; dess syfte är endast att hitta vissa grammatiska strukturer som finns specificerade i en regelfil, inte att bygga fullständiga parsningar utifrån alla tokens. Denna egenskap hos taggningen (att endast tokens som matchat en regel får en PoS-annotering) utnyttjas vid bakåtreferenser i parsningsmotorn. Exempelutdrag ur taggningsfiler återfinns i bilagorna C och G.

3.3 Frasinformationsfilen

Frasinformationsfilen (*span.xml*) innehåller information om de olika spann som sträcker sig genom texten. Ett spann är alltså de grammatiska strukturer man är intresserad av att annotera. Taggarna i filen heter *span* och har inte mindre än fem attribut. Det första är, som för de andra annoteringstaggarna, *id*. Nästföljande två attribut är *from* och *to*, vars värden sätts till de id som finns i tokenfilen. Attributet *rule_node* anger regelnamnet på den sista regel man kom till, det vill säga det löv i trädet som orsakade att just detta spann taggades. Attributet *tag_name* anger slutligen värdet på regelns attribut *tagname* om sådant finns angivet. Detta värde styr vilken färg spannet kommer att visas med i webbgränssnittet (se avsnitt 7.2 på sidan 41). Ett exempel på en *span*-tagg:

```
<span id="span.0" from="token.0" to="token.1"
rule_node="pres_accord_participe" tag_name="c07"/>
```

Spann som sådana kan vara nästlade, det vill säga ett spann kan innesluta andra. Exempelvis kan ett yttre spann sträcka sig från token 1 till token 6 och två inre från token 1 till token 2 respektive från token 4 till token 5. Annoteringsförslaget visar ett sätt på vilket detta kan taggas, men vi har valt att endast beskriva de längsta, icke-överlappande spannen i vår frasinformationsfil. Detta att mindre spann inkluderas i större utnyttjas framförallt på två sätt i Direkt Profil:

[23] Detta kan ge upphov till en del märkliga annoteringar, vilket beskrivs i avsnitt 5.3 på sidan 34

1. När taggningen ska visas visuellt i webbgränssnittet görs detta med färger som ju inte kan överlappa varandra i utmatningsfönstret. För att annoteringen ska kunna visas på rätt sätt i webbgränssnittet måste alla tokens ingå i ett spann, därav de tomma spannen.

2. I regelfilen taggar vi mindre spann som kan komma att skrivas över av ett större spann längre ner i trädet. Detta görs för att undvika att vissa taggningar inte ska sträcka sig till fönstrets slut om det man söker inte hittas. Exempel: *j'ai un chien* ska parsas så att *j'ai* annoteras som *être/avoir* med kongruens men utan efterföljande particip. Orden *un chien* ska naturligtvis inte tas med i spannet. När man funnit att det råder kongruens mellan pronomenet och verbet söker man vidare efter ett particip inom tre ord framåt. Det är dock först när man sökt till fönstrets slut man kan konstatera att där inte fanns något particip, men då finns ingen möjlighet att gå tillbaka i tokenströmmen och tagga tokens som redan behandlats. Vi löste detta genom att redan då kongruens mellan *j* och *ai* konstaterats, tagga *j'ai* som *être/avoir* med kongruens men utan efterföljande particip. Skulle det sedan visa sig att man finner ett particip taggas hela spannet som *passé composé* med kongruens. Det är först vid participkontrollen som räknare räknas upp, så inga räknare behöver räknas ned om det upptäcks att man gjort en felaktig taggning. (Denna typ av lösning används i noderna 8, 16 och 31 i det träd som beskrivs i avnsitt 4.1 på nästa sida.)

Exempelutdrag ur frasinformationsfiler återfinns i bilagorna D och H.

3.4 Räknarfilen

För enkelhetens skull har vi samlat informationen från ovanstående tre filer i *merged.xml* som innehåller såväl *token-, tag-* och *span*-taggar som räknartaggarna från räknarfilen. De sistnämnda heter *counter* och har fyra attribut. Den första är som vanligt *id* och används för att referera till en viss *counter*-tagg. Den andra heter *counter_id* och anger värdet på räknarens *id*-attribut i regelfilen. Dess värde styr tillsammans med attributet *counter_name* hur räknaren presenteras i webbgränssnittet. Det sista attributet *value* anger räknarens värde, det vill säga hur många förekomster av den aktuella strukturen som hittades i den parsade texten. En *counter*-tagg ser typiskt ut så:

```
<counter id="counter.11" counter_id="c11"
counter_name="Être/Avoir accord sujet/verb" value="1"/>
```

Exempelutdrag ur räknarfiler återfinns i bilagorna E och I. Det är främst *merged.xml* som används vid testning. Den är dessutom det enklaste sättet att kontrollera resultatet från standaloneversionen av Direkt Profil (se avsnitt 7.1 på sidan 40).

Det annoteringsformat vi valt ger unika id till både tokena och annoteringstaggarna och innehåller därmed den information som behövs. Man påpekar dock själv i förslaget till annoteringsformatet att det är svårläsligt (för människor) och att det som främst försvårar läsningen är att annotering och text är separerade och man föreslår att informationen i filerna ska presenteras för den mänskliga användaren genom andra verktyg [3]. Det är just vad som görs i webbversionen av Direkt Profil (se avsnitt 7.2 på sidan 41).

4 Regelfilen

De tidigare kapitlen har visat vad hur den typiska inputen till Direkt Profil ser ut, hur den skall behandlas med hjälp av trädet samt hur den önskade outputen ser ut. Nu återstår att visa hur detta kan automatiseras i systemet. Denna automatisering sker genom de inre mekanismerna i Direkt Profil och kan sägas bestå av tre samverkande huvuddelar nämligen reglerna, lexikonet och parsningsmotorn. Detta kapitel beskriver reglerna, nästa kapitel behandlar lexikonet och därpå följer en teknisk översikt över hur parsningsmotorn arbetar. Det har funnits flera olika versioner av regelfiler allteftersom fler noder implementerats och testats. Den senaste av dessa filer, *rules14.xml*, implementerar hela trädet.

4.1 Skillnaderna mellan beslutsträdet och regelfilen

Om man implementerar reglerna på det sätt som anges i beslutsträdet kontrolleras kongruensen mot det senaste token som matchade en regel och därmed PoS-taggades. I satsen *j'ai mangé* kommer man, efter att particip konstaterats, söka kontrollera kongruens mellan *mangé* och *ai* istället för mellan *ai* och *j*. Detta eftersom det inte finns någon möjlighet att ange vilka token en kongruenskontroll avser; de antas alltid vara det aktuella token man befinner sig på och det senast PoS-taggade.

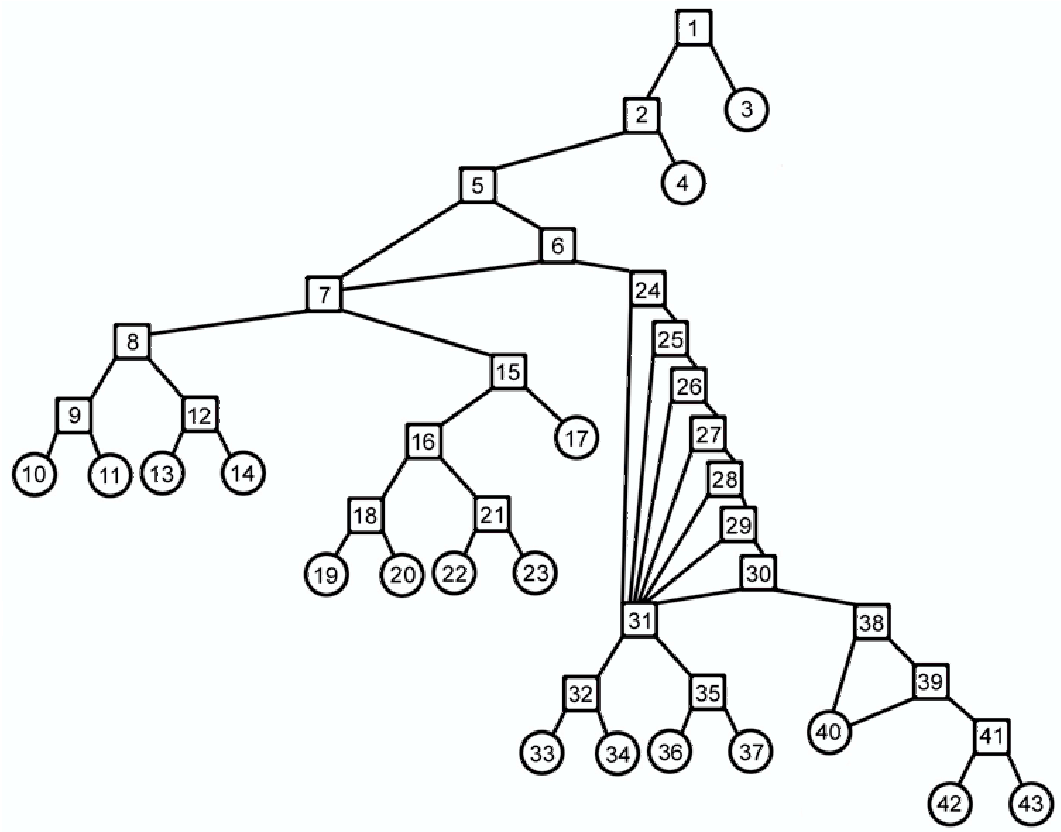
Problemet med att man inte kan ange mellan vilka noder en kongruenskontroll ska ske kringgås genom att vi byter plats på noderna i trädet i förekommande fall. Detta innebär att nod 4b fått byta plats med nod 4d och 4g (nod 8-9 och 12 nedan) och att nod 4l och de två noderna 4n bytt plats (nod 16, 18 och 21 nedan). Dessutom har nod 5a bytt plats med 5c och 5f.

Vidare har vi delat upp en del noder som kontrollerar flera olika lemman. Detta berör nod 4 som har delats upp i nod 5-6 i trädet nedan och nod 5 som har delats upp i noderna 24-30 i trädet nedan. Detta eftersom de kontrollerade olika lemman i samma nod i orginalträdet. I vår tillämpning har vi däremot endast kunnat kontrollera ett lemma i varje nod. Även nod 6a har delats upp och blivit noderna 38-39 nedan, samtidigt som ja/nej-svaret inverterats. Det beror på att nod 6a i orginalträdet frågade om verbet var personböjt. Vi har valt att definiera det som att de är personböjda i alla fall utom då de står i infinitiv eller particip. Ett ja-svar på frågan gjorde att man gick till vänster i orginalträdet. Nu frågar man istället om verbet står i infinitiv respektive particip och går därför till höger för att ange att verbet är personböjt.

Slutligen har en del noder strukits eftersom de inte ställde några frågor. I de fall de räknade upp räknare var de lika med summan av två efterkommande räknare, så ingen information har i och med detta gått förlorad (se diskussion i avsnitt 4.7 på sidan 29). Detta berör noderna 4c, 4k, 4m, 5b, 6 och 7. Reglerna implementerar alltså ett träd som ser ut som följer:

1. Är ordet ett pronomen[24]?
2. Leta upp första förekomst av verb inom fem ord framåt.
3. Gå till nästa ord.
4. Räkna upp en förekomst av sats utan verb.
5. Är verbet en form av *avoir*?
6. Är verbet en form av *être*?
7. Är verbet böjt i presens?

[24]Detta är en mindre specifik fråga än den som ställdes i ursprungsträdet. Att vi ställer en så pass vid fråga här beror på att information angående om ett pronomen fungerar som subjekts- eller objektspronomen saknas i lexikonet. Se vidare i avsnitt 5.3 på sidan 33



Figur 2: Trädet efter anpassningen till regelformalismen.

8. Råder kongruens mellan pronomenet och *être/avoir*-verbet?
9. Finns det ett verb i particip inom tre ord framåt?
10. Räkna upp en förekomst av *passé composé* i presens med kongruens.
11. Räkna upp en förekomst av *être/avoir* i presens med kongruens.
12. Finns det ett verb i particip inom tre ord framåt?
13. Räkna upp en förekomst av *passé composé* i presens utan kongruens.
14. Räkna upp en förekomst av *être/avoir* i presens utan kongruens.
15. Är verbet böjt i *imparfait*?
16. Räkna upp en förekomst av *imparfait*. Råder kongruens mellan pronomenet och *être/avoir*-verbet?
17. Räkna upp en förekomst av verb i andra tempus/modus.
18. Finns det ett verb i particip inom tre ord framåt?
19. Räkna upp en förekomst av *passé composé* i presens med kongruens.
20. Räkna upp en förekomst av *être/avoir* i presens med kongruens.
21. Finns det ett verb i particip inom tre ord framåt?
22. Räkna upp en förekomst av *passé composé* i presens utan kongruens.
23. Räkna upp en förekomst av *être/avoir* i presens utan kongruens.
24. Är verbet en form av lemmat *vouloir*?
25. Är verbet en form av lemmat *pouvoir*?
26. Är verbet en form av lemmat *savoir*?

27. Är verbet en form av lemmat *devoir*?
28. Är verbet en form av lemmat *faire*?
29. Är verbet en form av lemmat *laisser*?
30. Är verbet en form av lemmat *falloir*?
31. Råder kongruens mellan pronomenet och hjälpverbet?
32. Finns det ett verb i infinitiv inom tre ord framåt?
33. Räkna upp en förekomst av modalt hjälpverb med kongruens + infinitiv.
34. Räkna upp en förekomst av modalt hjälpverb med kongruens.
35. Finns det ett verb i infinitiv inom tre ord framåt?
36. Räkna upp en förekomst av modalt hjälpverb utan kongruens + infinitiv.
37. Räkna upp en förekomst av modalt hjälpverb utan kongruens.
38. Står verbet i infinitiv?
39. Står verbet i particip?
40. Räkna upp en förekomst av icke-personböjt lexikalt verb.
41. Råder kongruens mellan pronomenet och verbet?
42. Räkna upp en förekomst av lexikalt verb med kongruens.
43. Räkna upp en förekomst av lexikalt verb utan kongruens.

Dessa omkastningar av noder fungerar bra i det beslutsträd vi använt, men det kan inte uteslutas att denna lösning kommer att visa sig otillräcklig i framtiden, med andra beslutsträd. Ett exempel kan vara om man vill kontrollera kongruens mellan verben och pronominet i meningen *Elles sont intéressées.* När man kommer till participet vill man kontrollera om det har samma genus och numerus som pronominet, men med den algoritm vi tillämpar i nuläget är detta omöjligt eftersom man inte kunnat markera pronominet med någon flagga som talar om att det är detta token man vill gå tillbaka och kontrollera mot i senare regler. Detta problem kan heller inte lösas genom att byta plats på noder i trädet, utan formalismen måste utökas för att tillåta flaggning av de ord man vill kontrollera. (En sådan utökning medför att även förändringar i motorn måste göras.)

4.2 Regelformalismen

Beslutsträdet har implementerats i form av regler i ett egendefinierat XML-format. De regler som används i den nuvarande versionen av Direkt Profil är en utbyggd variant av Kostadinov/Thulins formalism [17]. Reglerna var skrivna i XML och vi valde att behålla det så eftersom reglerna ska vara separerade från själva parsningsmotorn.

Tack vare att reglerna är skrivna i XML kan vi välja teckenuppsättningen ISO-8859-1, vilket förhindrar att franska specialtecken förvanskas när man flyttar regelfilen mellan olika plattformar. Vi hade problem först med att *être* inte förstods av Mac-datorer då teckenkodningenen var satt till UTF-8. I princip har varje nod mappats till en regel (med vissa undantag där regelformalismen inte varit tillräcklig för att implementera trädet sådant det såg ut i Schlyters och Granfeldts tappning. Skälet till att beskriva reglerna i en egen fil separat från parsningsmotorn var att det enkelt ska gå att anpassa Direkt Profil även för andra beslutsträd eller till och med för andra språk. Tanken är ju att systemet på sikt även ska kunna användas för nivåbestämning inom andra språk, exempelvis tyska. Regelformalismen bygger i stort på Kostadinov/Thulins arbete [17].

Det finns fyra olika typer av regeltaggar: räknarna, två olika tokeniseringsregler och så parsningsreglerna. En parsningsregel består av en regeltagg som har flera barntaggar, varav de

viktigaste är *search* och *action*. De flesta taggar har dessutom ett antal attribut som anger olika parametrar för just den taggen. Nedan visas de taggar som regelfilen är uppbyggd av, med ett indrag för varje taggnivå och därpå följer en mer ingående beskrivning av de viktigaste taggarna.

- rules
  - counters
    - counter
  - wordTokenizeRule
    - regex
  - sentenceTokenizeRule
    - regex
  - rule
    - description
    - example
      - ex_found
      - ex_notfound
    - search
      - inflection
        - gender
        - number
        - person
        - mode
      - lemma
      - regex
      - recursive
    - action
      - found
      - notfound

4.3 Counters

Counters-taggen innehåller ett antal *counter*-taggar som beskriver de räknare som finns i beslutsträdet. Räknarna behövs för nivåbedömningen av texterna; man vill räkna antalet förekomster av varje fenomen som är typiskt för en viss nivå. En *counter* har två attribut, *id* och *name*. Värdet på *id* är unikt för varje räknare och uppbyggt på ett strikt sätt för att kunna förstås vid visualiseringen i webbgränssnittet. Värdet på *name* används också i webbgränssnittet, nämligen i samband med att räknarnas värden presenteras (se avsnitt 7.2 på sidan 41). I den nuvarande versionen av reglerna är en *counter* inte kopplad till någon regel, utan kan räknas upp av flera regler. Detta möjliggör att de räknare som utgörs av summor av andra räknare kan räknas upp på flera ställen, exempelvis räknare 6b. Under utvecklingsarbetets gång har vi även provat att koppla varje räknare till en viss regel genom ett attribut *rule_id*, eftersom det underlättade behandlingen av räknarna i motorn. Denna design förkastades dock eftersom den omöjliggjorde att räknare kunde räknas upp på flera olika ställen och därmed utgöra summor.

4.4 Tokeniseringsreglerna

Tokeniseringsreglerna innehåller vardera exakt en *regex*-tagg som specificerar vilket reguljärt uttryck som ska fungera som sats- respektive ordavgränsare. Den första tokeniseringsregeln är

sentenceTokenizer. Här anges vilka tecken som ska anses vara satsavgränsande. Punkt, utropstecken, frågetecken, semikolon och kolon anses alla vara satsavgränsare.

Kommatecken har inte definierats som satsavgränsare helt enkelt därför att det inte är det. Hur nybörjare i franska använder kommatecken bör undersökas empiriskt innan det kan bli aktuellt att ta med det som satsavgränsare i Direkt Profil. Kommatecken förekommer ofta direkt efter adverbial (exempelvis i meningen *Après, nous allons au café*) och även precis som i svenskan i uppräkningar. Det har dock varit en svår avvägning, eftersom vårt val att inte definiera kommatecken som satsavgränsare kan medföra felaktiga parsningar. Detta kan exemplifieras genom meningen *Il, je crois, est un idiot.* Här kommer kongruens att kontrolleras mellan *il* och *crios* istället för mellan *je* och *crios* som hade blivit fallet om en ny sats ansetts börja efter kommatecknet[25]. Man kan naturligtvis komma runt detta problem även på andra sätt än att definiera kommatecken som satsavgränsare, vilket föreslås i avsnitt 10.1 på sidan 51.

Normalt sett i datalingvistiska sammanhang brukar punkt som satsavgränsare medföra vissa problem eftersom det används i många förkortningar. I Direkt Profil har detta inte orsakat några problem eftersom det knappast förekommer förkortningar i den här typen av texter. Punkt är därför en av de säkraste indikatorerna på satsslut.

Vi har även diskuterat möjligheten att definiera vissa ord som satsavgränsare. Om exempelvis *que* anses vara satsavgränsande skulle satsen *Il veut que je parler* inte felaktigt parsas som modalt hjälpverb med efterföljande infinitiv (som det gör i nuläget), utan som modalt verb samt icke-personböjt lexikalt verb (vilket är det önskvärda). De ord som eventuellt skulle vara lämpliga satsavgränsare är *que, qui, quand, si, où, et* och *mais*. Det måste dock testas empiriskt vilka av dessa som förbättrar Direkt Profils resultat. Speciellt *qui* är tveksam; vi har nämligen diskuterat att använda *qui* som subjektspronomen.

Hur ord ska avgränsas definieras i *wordTokenizeRule*. Inte heller detta har medfört några problem i våra tester av Direkt Profil. Vi använder helt enkelt $\backslash b$[26] som ordavgränsare.

4.5 Rule

Parsningsreglerna finns beskrivna i *rule*-taggarna. En rule är uppdelad i två delar som kallas *search* och *action*. I *search* anges vilken struktur man letar efter i denna regel och i *action* anges vad som ska hända om den påträffas respektive inte påträffas. Idén att dela upp reglerna i två delar (i vårt fall *search* och *action*) är ingalunda unik; man har inom Granska utvecklat ett eget regelspråk där de två regeldelarna (motsvarande *search* och *action*) markeras med en särskiljande pil. Granska bygger dock inte på ett beslutsträd på samma sätt som Direkt Profil och man har alltså ingen uppdelning mellan *found* och *notfound* (den senare saknas) och inte heller någon möjlighet att specificera en *nextrule* [4]. Även grammatikkontrollen i Word 97 använder sig av en struktur där en så kallad *descriptor*-regel består av en beskrivning till vänster om en implikationspil och en *action*-del till höger om densamma [12].

En rule har två attribut, *id* och *framesize*. Värdet på *id* är unikt för varje parsningsregel och används dels för att kunna särskilja dem och dels för att kunna hänvisa till dem. *Framesize* är fönsterstorleken som anges i beslutsträdet. Observera att ordet “fönster” här används metaforiskt för att beteckna ett avgränsat område som man har under behandling. Ordvalet är något okonventionellt på svenska, men det är vanligt förekommande i de stora europeiska språken (exempelvis *window* och *fenêtre*) . Mer specifikt betecknar “fönster” i den här rapporten de ord

[25] Med parsningsmotorns nuvarande implementation gör det inte någon skillnad om man anger kommatecken som satsavgränsare eller ej, se avsnitt 6.2 på sidan 38.

[26] Meta-tecken för en så kallad *word boundary*, för mer information om detta se http://www.regular-expressions.info/wordboundaries.html

man just nu söker igenom i den aktuella satsen. Detta har kallats *framesize* på engelska när vi skulle välja attributnamn till fönsterstorleksparametern.

Vilket värde fönsterstorleken ska sättas till är inte något enkelt val och det finns inget självklart optimalt värde. Värdet kan behöva varieras för olika typer av texter. De värden på fönsterstorlekar vi valt, nämligen att söka efter ett verb inom fem ord från pronomenet och att söka efter ett particip inom tre ord från *être/avoir*, tycks enligt språkinlärningsforskningen vara lämpliga mått. Dessa värden bör dock testas empiriskt och kan komma att ändras.

En för liten fönsterstorlek å ena sidan kan resultera i att ord inte hittas trots att meningen är korrekt uppbyggd. Satsen *nous, les filles de Suéde, voulons...* kommer att parsas som Mening utan verb, eftersom fönsterstorleken är fem och verbet *voulons* är sjunde tokenet ordet efter pronomenet *nous*.

En för stor fönsterstorlek å andra sidan kan leda till att ord parsas som om de tillhörde en struktur trots att de inte gör det. Ett exempel är följande meningar: *Ils prennent un café. Elle aussi mais moi je prends un croissant.* Med en fönsterstorlek på fem ord efter påträffat *elle* kommer man att söka efter ett verb och alltså ignorera eventuella nya pronomen. *Elle* kommer därför att matchas med *prends* trots att detta tillhör *je*.

Framesize anger hur många ord som ska undersökas framöver innan man anser att det man söker efter inte påträffats. En *rule* appliceras på nästföljande token efter det som förra regeln applicerades på, förutom i de fall man har regelkedjat[27]. Varje parsningsregel innehåller fyra barntaggar. De första två, *description* och *example* används endast i användargränssnittet i webbversionen av Direkt Profil. Innehållet i dessa taggar är en beskrivning av vad regeln gör, respektive exempel på regelns tillämpande. De sista barntaggarna, *search* och *action* utgör parsningsregelns två huvuddelar. De har därför ägnats egna avsnitt nedan.

4.6 Search

Taggen *search* har inte några attribut. Den måste dock innehålla minst en barntagg. Möjliga barntaggar är *inflection, lemma, accord, recursive* och *regex*. Den sistnämnda fanns med i Kostadinov/Thulins regelformalism och vi har inte sett något skäl att ta bort den, trots att varken vi eller Kostadinov/Thulin använder den. Den är alltså helt otestad. Sökning efter ord sker efter tre möjliga kriterier[28]:

1. Sök efter en viss ordklass, eventuellt med en viss böjning (vissa särdrag).
2. Sök efter ett visst lemma.
3. Sök efter kongruens mellan två ord.

Inflection är den tagg som används för den första av tre möjliga sökkriterierna. Taggen kan, som namnet antyder, användas för att söka efter en viss böjning av ett ord. Vilka olika böjningar som är möjliga att ange beror på vilken ordklass man söker efter. Ordklassen anges med hjälp av attributet *category* som kan anta värdena *pronoun, verb, noun, adjective, int_pronoun, determiner, adverb, preposition, conjunction, numeral, interjection, abbrevation* och *residual*. Endast *pronoun* och *verb* används hittils i Direkt Profil. Den önskade böjningen anges genom de fem barntaggarna *gender, number, person, tense* och *mode*. Beroende på vilken ordklass som angetts kan olika kombinationer av barntaggarna förekomma; för pronomen anges inte tempus (*tense*)

[27] Med regelkedjning förstås här att *framesize*-attributet i *rule* är satt till noll samt att *search* har barntaggen *recursive*. Se avsnitt 4.6 på sidan 29.

[28] Användningen av ordet kriterium här ska inte sammanblandas med taggen *criterium* som används som barnbarntagg i search.

eller modus (*mode*) och det är endast verb i particip som kan ha genus (*gender*) angivet. Ett exempel på hur inflection används ges här:

```
<search>
   <inflection category="verb">
      <tense value="past"/>
      <mode value="participle"/>
   </inflection>
</search>
```

Detta är sökdelen i parsningsregeln som motsvarar nod 39, alltså sökning efter ett verb i particip. De möjliga värden som *value*-attributen av inflections barntaggar kan anta återfinns i bilaga J. Taggen kan också användas utan barntaggar för att bara söka efter en viss ordklass, vilket utnyttjas bland annat i trädets första regel där man söker ett pronomen, vilket som helst:

```
<search>
    <inflection category="pronoun" />
</search>
```

Lemma-taggen möjliggör det andra sökkriteriet i search. Denna tagg används för att söka efter ord med ett visst lemma[29] och har inga barntaggar. Det enda som finns mellan start- och sluttagg är det sökta lemmat. *Lemma*-taggen används i vårt beslutsträd i nod 6 för att kontrollera om det verb man funnit är en form av *être*:

```
<search>
   <inflection>
      <lemma>être</lemma>
   </inflection>
</search>
```

Accord är den tagg som används för det tredje och sista sökkriteriet[30]. Taggen används för att kontrollera kongruens mellan två ord och har tre barntaggar. Namnet på den första av dessa är *criteriums* och de två sista *category*. *Criteriums* innehåller i sin tur minst en *criterium*-tagg som anger vilka särdrag kongruenskontrollen ska ske utifrån. Av de två *category*-taggarna anger den första vilken ordklass det aktuella tokenet[31] förväntas ha och den andra vilken ordklass föregående token[32] förväntas ha.

```
<search>
   <accord>
      <criteriums>
         <criterium value="number"/>
         <criterium value="person"/>
     </criteriums>
     <category value="verb"/>
     <category value="pronoun"/>
   </accord>
</search>
```

[29] Se ordlistan på sidan 57.

[30] Det svenska ordet kongruens översätts närmast med *agreement* på engelska, men vi har valt det franska ordet för kongruens, nämligen *accord*.

[31] Det token som denna regel appliceras på.

[32] Det senast PoS-taggade tokenet.

Kostadinov/Thulins regelformalism har utvecklats med flera olika taggar. Beskrivningen ovan gäller den regelformalism som Direkt Profil implementerar i nuläget. En del taggar som vi lagt till är mest för ökad läslighets skull, till exempel *description* och *example*. Andra står för viktig funktionalitet, till exempel *accord* och *criteriums*. I bilaga K listas alla taggar som vi utökat regelformalismen med.

Recursive[33] är den sista barntaggen till *search* och används endast för att markera regelkedjning, det vill säga att sökningen ska ske på samma ord som man stod i när förra sökningen gjordes. Regelkedjning markeras på två sätt i vår regelformalism, dels genom att fönsterstorleken är satt till noll och dels genom taggen *recursive* i *search*-delen.

4.7 Action

Taggen *action* har inte några attribut. Den måste dock innehålla två barntaggar, *found* och *notfound*. Den förstnämnda specificerar vad som ska göras om det man söker efter[34] påträffats och den andra vad som ska göras om man inte, inom den angivna fönsterstorleken, funnit det man söker. *Found* och *notfound* har inga barntaggar i sin tur, men ett antal attribut. Dessa används för att närmare specificera om några av följande tre åtgärder ska vidtas:

1. Räkna upp en räknare.
2. Tagga aktuellt fönster.
3. Gå vidare till nästa regel.

Inccounter är det attribut som anger vilken räknare som ska räknas upp. Dess värde sätts till räknarens id (vilket angavs som attribut i motsvarande *counter*-tagg). Räknaren räknas då upp med ett. Endast en räknare kan räknas upp, varför de räknare som ska räknas upp samtidigt som andra (4c, 4m, 5b och 7) inte räknas upp. De är ju summor av andra räknare (4c=4e+4f, 4m=4o+4p, 5b=5g+5h, 7=6d+6e). I framtida versioner av Direkt Profil skall denna brist i regelformalismen åtgärdas, vilket enkelt kan göras genom att *incconter* blir barntagg istället för attribut till *found* respektive *notfound*. Skälet till att detta ännu inte gjorts är att behandlingen av räknarna i motorn är implementerad så att endast en räknare per *found/notfound* tillåts. Att åtgärda detta låter sig inte göras lika enkelt.

Tagname är det attribut som anger vilket taggnamn det aktuella fönstret ska annoteras med. Det hamnar i *span.xml* (se exempel i bilagorna D och H) och visas dessutom i resultatfönstret i standalone-versionen av Direkt Profil (se avsnitt 7.1 på sidan 40). Detta attribut förekommer tillsammans med *dotagging* som anger om det aktuella fönstret ska taggas eller ej. Attributet *dotagging* kan anta värdena *yes* eller *no*.

Nextrule är det attribut som anger vilken parsningsregel man ska gå vidare till. Ett *nextrule*-attribut finns alltid angivet, både i *found*- och *notfound*-taggarna, förutom i de regler som motsvarar löv i trädet. *Found*-taggen anger vad som händer då man svarat ja på den fråga som ställs i noden (alltså i *search*) och därmed går till vänster nedåt i trädet. *Notfound* motsvarar på samma sätt att man går till höger i trädet. Om ingen *nextrule* finns angiven anropas helt enkelt begynnelseregeln igen. Den har hårdkodats in som *rule1*.

[33] Benämningen *recursive* är kanske inte ett så lyckat namn eftersom det ger andra associationer än vad som avses här. För en datalog för det närmast tankarna till algoritmernas återupprepande av exekveringsgång med ny indata. Här är det tvärtom; samma indata (samma ord) återupprepas, med en ny exekveringsgång (en ny regel). För en lingvist kanske användningen av ordet *recursive* ger associationer till nästlade satser, till exempel förekomst av en nominalfras inuti en annan nominalfras. Eftersom *recursive*-taggen är överflödig är den planerad att tas bort i framtida versioner av formalismen och något nytt namn har inte ansetts nödvändigt.

[34] Det som angetts i search.

Nedan visas ett exempel på hur *found* och *notfound* används. *Found* beskriver nod 15, det vill säga att den funna strukturen ännu inte ska taggas, men *imparfait*-räknaren (räknare 4k) ska räknas upp och man ska gå vidare ner i trädet till regel “impf_accord_etre_avoir” som motsvarar nod 16 i det träd vi implementerar. *Notfound* beskriver nod 4s, det vill säga att man inte funnit *imparfait* inom det givna fönstret, varför det taggas och räknare 4s räknas upp. Detta är ett löv och därför finns ingen *nextrule* angiven.

```
<action>
   <found inccounter="c17" nextrule="impf_accord_etre_avoir"
                                          dotagging="no"/>
   <notfound inccounter="c22" dotagging="yes" tagname="c22"/>
</action>
```

5 Lexikonet

Det lexikon vi använder kommer från ABU[35]-CNAM[36]. Det finns att ladda ner från webbplatsen http://abu.cnam.fr/DICO och är gratis att använda för icke-kommersiella syften. Lexikonet består av nästan 290.000 uppslagsord (*entries*), men det är inte uppbyggt som ett vanligt "boklexikon". De brukar nämligen vara lemmatiserade, vilket innebär att orden endast står listade i sin grundform och därför kan man inte slå upp böjningar av ord. Om man vill hitta ordet *suis* i ett vanligt boklexikon måste man därför veta att det är en böjd form av *être*. *Être* är ordets lemma[37] och *suis* en av dess böjningar. I ABU-CNAMs lexikon däremot finns *suis* med som ett eget uppslagsord (*entry*). Därför innehåller inte lexikonet 290.000 olika ord, även om det finns så många uppslagsord. Antalet olika ord, det vill säga antalet lemman, är knappt 55.000. Det finns möjlighet att använda lemmatiserade lexikon även i datalingvistiska sammanhang; inom projektet Granska använder man sig av böjningsregler som appliceras på lemmana. Därigenom kan alla möjliga böjningsformer av orden härledas. Den stora fördelen med detta är uppenbarligen att lexikonstorleken minskar [4]. Lexikonstorleken är en viktig fråga i kommersiella program och man har därför varit tvungen att reducera storleken på det ursprungliga lexikonet i Word 97 [12]. I Direkt Profil har sådana hänsyn inte tagits.

5.1 Något om lexikonets uppbyggnad

Lexikonet är uppbyggt så att varje rad innehåller ett uppslagsord med information om dess lemma och PoS. Raden med uppslagsordet *suis* ser ut så:

```
suis    être    Ver:IPre+SG+P1
```

En rad i lexikonet består alltså av tre enheter; först själva uppslagsordet, därefter dess lemma och sist PoS-informationen som i sin tur är uppdelad i mindre enheter åtskilda av kolon. Den kod som står innan kolonet är ordklassen. ABU-CNAM använder följande koder för ordklasserna:

Ordklass	*Kod*
Verb	Ver
Substantiv	Nom
Adjektiv	Adj
PrePoSition	Pre
Pronomen	Pro
Interjektion	Int
Determinant	Det
Konjunktion	Con
Förkortning	Abr
Adverb	Adv

Det som står efter kolonet är särdragen[38]. Ambiga[39] uppslagsord har flera möjligheter när det gäller särdragen, också de är åtskilda av kolon i lexikonet. Beroende på vilken ordklass det rör sig om ser särdragskodningen lite olika ut. Vi använder endast pronomen och verb i Direkt Profil så jag beskriver endast särdragskodningens utseenden för dem. För pronomina finns tre olika möjligheter. Observera att det finns två olika sätt att ange person:

[35] Association des Bibliophiles Universels
[36] Conservatoire National des Arts et Métiers
[37] Se ordlistan på sidan 57.
[38] Se ordlistan på sidan 58.
[39] Se ordlistan på sidan 56.

Numerus+Person
Genus+Numerus
PersGenus+Numerus

Variablerna i dessa tre möjliga scheman kan anta följande värden:

Variabel	*Kod*	*Betydelse*
Numerus	PL	Pluralis
	SG	Singularis
Person	P1	Första person
	P2	Andra person
	P3	Tredje person
Genus	Fem	Femininum
	Mas	Maskulinum
	InvGen	Både Mas och Fem
Pers	1	Första person
	2	Andra person
	3	Tredje person

Verbens särdragskodning följer följande två scheman:

ModusTempus+Numerus+Person
ModusTempus+Genus+Numerus

Numerus, Person och Genus kan anta samma värden som för pronomen. De andra två variablerna kan anta följande värden:

Variabel	*Kod*	*Betydelse*
Modus	C	Konditionalis
	I	Indikativ
	S	Konjunktiv
	Im	Imperativ
	P	Particip
Tempus	Pre	Presens
	Imp	*Imperfait*[40]-konstruktion
	Pas	Particip
	PSIM	*Passé Simple*-konstruktion
	Fut	Futurum

Infinitivformer av verben kodas dock endast med ordklass och infinitivmarkör:

```
être   être   Ver:Inf
```

Detta exempel betyder alltså att uppslagsordet *être* har lemmat *être* och är ett verb i infinitiv, medan det tidigare exemplet med *suis* betyder att *suis* också har lemmat *être* och är ett verb i indikativ, presens, första person, singularis.

5.2 MiniDict, dico och MediumDict

Lexikonet distribueras i form av 26 textfiler, en för varje begynnelsebokstav hos lemmana. Det innebär att även om de böjda formerna, uppslagsorden, börjar på en annan boksatav än sitt

lemma hamnar de ändå i den fil som lemmat tillhör. (Exempelvis finns *suis* i *E.txt* istället för *S.txt*) Dessa 26 filer hade satts ihop av Kostadinov/Thulin till en enda stor fil, kallad dico. Den innehåller alltså nästan 290.000 rader och är därför "tung" att arbeta med. Som komplement har vi i utvecklingsarbetet av Direkt Profil använt oss av en mindre fil vid namn MiniDict. Den är på drygt 100 rader och innehåller de uppslagsord som vi behövt för just de texter vi utvecklat systemet med. Den har skapats genom att vi helt enkelt kopierat de rader vi behövde ur dico.

Vid körning med autentiska texter måste dico användas för att kunna hitta orden i lexikonet. Fördelen med dico är att den innehåller de flesta ord som kan tänkas behövas. Fördelen med MiniDict är att den inte innehåller några "onödiga" ord, som tar upp resurser. Vi har diskuterat att kombinera dessa fördelar i ett lexikon med de ca 2000 vanligaste orden i franskan. Lexikonet, som går under arbetsnamnet MediumDict, skulle förmodligen vara tillräckligt för att parsa de texter som är aktuella för Direkt Profil. Språkinlärare har nämligen en tendens att ha ett ganska begränsat (men gemensamt) ordförråd i början av inlärningsprocessen och detta beräknas bestå av ett par tusen ord. MediumDict skulle kunna skapas på liknande sätt som MiniDict, det vill säga genom att utifrån en listning över de vanligaste orden kopiera raderna för de lemmana från dico. Någon sådan listning finns dock inte att tillgå på Romanska Institutionen och vi har inte funnit någon annorstädes heller.

Vi har gjort försök att skapa en MediumDict genom att endast kopiera över pronomina och verb från dico. Det skulle inte medföra några inskränkningar i funktionaliteten hos dagens Direkt Profil eftersom endast pronomina och verb parsas. Detta reducerade dock inte storleken så mycket som vi hoppats; antalet uppslagsord som är verb eller pronomen är över 190.000. (Antalet pronomenuppslagsord är försvinnande litet, endast lite drygt 100.) Det kan jämföras med antalet uppslagsord tillhörande ordklassen substantiv som var lite drygt 60.000. Om man inte tar med alla böjningsformer av verben skulle storleken kunna reduceras eftersom storleksförhållandena är de omvända om man tittar på antalet lemman; knappt 40.000 substantiv medan verb och pronomen tillsammans inte överskrider 8.000 (varav endast ett 60-tal är pronomenlemman). Vi har valt att inte lägga tid på att välja ut vilka böjningsformer som skulle kunna komma ifråga, främst av tre anledningar:

1. Storleken på dico utgör inte något allvarligt problem, varken vid programstart, exekvering eller interaktion med Direkt Profil. Det är framför allt engångsinläsningen av lexikonet i början av programexekveringen som tar tid. Därefter sparas varje uppslagsord i form av ett objekt i en hashtabell, vilket gör att sökningen i hashtabellen går acceptabelt snabbt (se nedan i avsnitt 6.1 på sidan 37).

2. En rent syntaktisk reduktion av uppslagsorden i lexikonet skulle inte reducera storleken så mycket som önskat eftersom antalet verblemman är så pass stort. Det skulle krävas en semantisk reduktion, det vill säga att vissa ord togs bort på grund av deras låga frekvens.

3. I framtida versioner av Direkt Profil kan man komma att implementera beslutsträd som behandlar exempelvis substantiv. Då skulle den diskuterade MediumDict komma att vara otillräcklig.

5.3 Konsekvenser av lexikonets informationsinnehåll

Lexikonet är inte tillräckligt i det avseendet att det inte skiljer mellan olika sorters pronomen; varken förenade/oförenade eller subjekts-/objektspronomen särskiljs. Såväl *je* och *me* som *moi* har PoS-informationen Pro:SG+P1 i lexikonet. Det finns alltså ingen möjlighet att skilja dem åt. Att utöka lexikonet med den nödvändiga informationen för att kunna göra denna åtskillnad bedömdes dock falla utanför ramen för detta examensarbete.

Denna avsaknad av information kan dock ge besynnerliga följder. I utvecklingsarbetet parsades *Deux personne, une fille et sa mère, partent*. Ordet *personne* står med i lexikonet både som pronomen och substantiv. Eftersom parsningsmotorn väljer den möjlighet som bäst matchar regeln väljs den PoS-information som motsvarar pronomen. Inget verb följer dock inom fem ord så en förekomst av sats utan verb anses hittad. Samma fenomen kan också leda till att pronomen hoppas över, exempelvis i satsen *La seule langue que nous parlons bien est le suédois*. Pronomenet *nous* hoppas över eftersom *la* tolkats som pronomen. Verbet *parlons* matchas med *la*. För att undvika den här typen av fel tills den extra informationen om pronomina lagts till i lexikonet, har vi helt enkelt tagit bort de pronomen som orsakat problem. De listas i bilaga L. Dock skall åtminstone pronomina *eux, toi, moi* och *lui* återföras i lexikonet ganska snart. De kan nämligen fungera som subjektspronomen ibland, även om de sällan gör det på dessa inlärarnivåer.

En del fel kommer man aldrig ifrån, åtminstone inte med ett renodlat morfosyntaktiskt system[41]. Vi har funnit tre exempel på feltolkningar som uppstår till följd av avsaknad av semantisk information i systemet. De är så kallade ambiguitetsproblem[42] vilket innebär att en form kan ha flera olika betydelser. Alla tre ambiguitetsproblemen är konsekvenser av innehållet i lexikonet i kombination med att parsningsmotorn väljer den tolkning ur lexikonet som bäst matchar aktuell regel. Vi har tillfälligt löst dessa problem genom att plocka bort uppslagsord ur lexikonet, men förmodligen finns fler ord som bör plockas bort som vi ännu inte upptäckt. Vilka som ska bort kan vi bara finna på empirisk väg och detta är alltså inte en lämplig lösning på lång sikt. De ord som tagits bort ur lexikonet listas i bilaga M.

1. Substantiv tolkas som verb. Vissa vanliga substantiv är även böjningar av (inte lika frekventa) verb. Verbvarianten väljs alltid eftersom substantiv inte matchar några regler i trädet. Ett exempel är den inkorrekta inlärarmeningen *Je á Paris en voiture* som parsas som lexikalt verb med kongruens, eftersom *voiture* tolkas som verbet *voiturer* böjt i presens indikativ, första person singularis. Det vore önskvärt att Direkt Profil istället parsade detta som sats utan verb.

2. Fel verblemma väljs. Vissa böjningar av verb ser likadana ut för olika lemman. Första bästa lemma som matchar väljs, varför det som råkar stå först i lexikonet kommer att väljas om båda formerna har kongruens med pronomenet. Ett exempel är det relativt vanligt förekommande *ils étaient* vilket – på de språknivåer Direkt Profil arbetar med – i de allra flesta fall rör sig om *être/avoir* i *imparfait* med kongruens. Dock är formen *étaient* även en böjning av lemmat *étayer*, nämligen presens indikativ, tredje person pluralis. Den råkar stå först av de två möjligheterna i lexikonet, varför den väljs. Samma fenomen uppträder för formen *suis*, som är en böjd form av både *être* och *suivre*. I det fallet kommer dock *être*-varianten att väljas eftersom den står först i lexikonet.

3. Pronomen tolkas som verb. Vi har funnit en förekomst av ett uppslagsord som är både ett pronomen och ett verb, nämligen *tu*. I exemplet *j'étais lá mais tu non* kommer pronomenet *tu* att tolkas som verbet *taire* i particip, eftersom det matchar regeln i nod 9 som letar efter ett verb i particip inom tre ord efter det att hjälpverbet *étais* konstaterats. Detta kan lösas genom att man anger *mais* som satsavgränsare, se avsnitt 4.4 på sidan 25.

Många av ambiguitetsproblemen hade kunnat undvikas genom att lexikonstorleken minskas så att de mindre vanliga orden inte finns med.

[41] Ett system som alltså inte implementerar någon semantik.
[42] Se ordlistan på sidan 56.

5.4 Lexikonet i XML-format

I enlighet med våra tidigare designbeslut att så långt som möjligt använda XML-kodning för de språkspecifika delarna av Direkt Profil har även lexikonet konverterats till XML-format. Huvudskälet till att överföra lexikonet till XML-format är att man då kan ange vilken teckenuppsättning man vill ha. Om lexikonet sparas i UTF-8-formatet fungerar de franska tecknen inte problemfritt. I en XML-fil kan teckenuppsättningen anges till exempelvis ISO-8859-1. Denna standard fungerar på alla de plattformar vi testat (Unix, Windows, Mac).

Vi utarbetade ett rudimentärt XML-format som framförallt inspirerats av en föreslagen standard för morfosyntaktisk annotering som lagts fram inom ramen för RNIL[43] [7] [6]. Formatet ser ut som följer:

```
<wordform entry=UPPSLAGSORDET lemma=UPPSLAGSORDETS GRUNDFORM
pos=UPPSLAGSORDETS ORDKLASS features=UPPSLAGSORDETS BÖJNING(AR)/>
```

Möjliga värden är på *pos*-attributet är: *ver, nom, det, pre, pro, adj, adv, con, int, abr, qpro* och *ono*. Ännu återstår att formalisera det attribut som kallas *features*. Dess värde anger de olika särdrag som ordet kan motsvara och har bara överförts direkt från lexikonets textformat. Alla bokstäver har dock konverterats till små. Det innebär att i *features*-attributet kan det finnas flera olika möjliga böjningar. Detta kan representeras på två möjliga sätt vid en formalisering, men vi har inte tagit ställning till om någon av dem är att föredra:

1. Varje ord som har olika möjliga features förekommer flera gånger i filen.
2. Taggen *wordform* har *features* som barntagg istället för som attribut.

Verktyget för att konvertera lexikonet till XML-format är – liksom all annan programkod i Direkt Profil – skrivet i Java. Programmet kallas *changeDico* och läser in de 26 textfilerna en i taget och skriver ut den resulterande XML-koden i 26 nya filer. Vi behåller uppdelningen efter begynnelsebokstav för att filerna ska vara mer lättarbetade, bland annat för XML-validatorer som använts för att kontrollera att XML-koden är formellt korrekt. I samband med att lexikonet överförts till XML-format har vissa felaktigheter upptäckts och rättats till:

1. Det fanns nästan 300 dubletter i lexikonfilerna. De överfördes endast en gång till XML-formatet.
2. Ordklassen *con* stod som *conj* på ett ställe. Den skrevs som *con* till XML-filen.
3. Uniformering av pronomenkodning: De olika pronominas PoS-annotering finns i tre olika format i lexikonet (se ovan i avsnitt 5.1 på sidan 31). I XML-filerna står alla på formen Numerus+Person+Genus[44]. En del pronomina i lexikonet saknade information om person, andra om genus. De uppslagsord som var kodade på formen Pro:Genus+Numerus var alla i tredje person, varför denna information hårdkodades in i *changeDico*. Vid en eventuell vidareutveckling med ett annat lexikon är det alltså viktigt att man kontrollerar att detta antagande fortfarande stämmer innan *changeDico* används. Lemmana till de uppslagsord som var kodade på formen Pro:Genus+Numerus finns listade i bilaga N.

[43] Ressources Normalisées en Ingénierie de la Langue

[44] Möjligen behöver samma sak göras med vissa lemman som är kodade som “Det”. Det används inte i vårt system, men kan komma att implementeras i framtiden.

4. En del pronomen saknade information om ordklass, så denna information lades till. Detta berör uppslagsord med följande lemman: *ça, celles-ci, celle-ci, celui-ci, ceux-ci, elle-même, lui-même, moi-même, nous-mêmes* och *soi-même*.

5. Rader som saknade uppslagsord hoppades också över. De berörda raderna finns listade i bilaga O.

6. En rad med uppslagsordet *ut* och lemmat *un* har tagits bort eftersom den verkade felaktig.

7. Raden "ce ce Pro:SG+P3PL+P3" verkar ha en formellt felaktig PoS-annotering och har hoppats över.

8. Lemmana *défonc* och *révo* verkar inte finnas i franska språket, åtminstone inte i TLF[45] och har tagits bort. Följden blev att tre uppslagsord togs bort ur lexikonet.

9. Uppslagsorden med lemma *icelui* (fyra stycken) har tagits bort eftesom de är ålderdomliga och om de skulle förekomma i en text som Direkt Profil parsar är det mer troligt att det är en felstavning av exempelvis *celui*.

10. Imperativ verkar anges på två olika sätt i lexikonet. Exempelvis var två av uppmaningsformerna av vouloir kodade på olika former:
    veuillez vouloir Ver:ImPre+PL+P2
    veuillons vouloir Ver:Imp+PL+P1
    Formen ImPre förekom endast på tre ställen i lexikonet och antogs vara felaktig. För att undvika sammanblandning med imperfekt, som kodas på formen ModusTempus, valde vi att imperativ genomgående skulle kodas på formen Modus+Num+Pers, där Modus = "Im"för imperativ.

[45] Le Trésor de la Langue Francaise, sökning på webplatsen http://atilf.atilf.fr/tlf.htm den 14 juni 2004

6 Parsningsmotorn

Parsningsmotorns uppgift är att läsa in ett lexikon och ett antal lingvistiska regler som finns definierade i XML-format och sedan applicera reglerna på en inputtext. Den nuvarande versionen av motorn är en avknoppning av den äldre och har samma grundstomme. Därför beskrivs nedan den äldre motorns struktur och sedan de utökningar som gjorts i den nya motorn. I båda avsnitten beskrivs de tekniska detaljerna ganska ingående. Detta är för att ge en djupare förståelse för den inre funktionaliteten än vad en översiktilig skissering över motorns uppbyggnad skulle ge. Dessutom kommer arbetet med Direkt Profil att fortskrida även efter detta examensarbetes avslutande och det är då önskvärt med ganska ingående förklaringar av de designval vi gjort.

6.1 Motorn i den äldre versionen

Kostadinov/Thulins parsningsmotor utgörs av ett 40-tal javaklasser. Systemet startas genom *MainProgram* som läser in lexikonet genom att anropa klassen *ImportDictController*. Här läses lexikonet in rad för rad och uppslagsorden sparas i objekt tillsammans med sina lemman och PoS-information. Objekten sparas i en hashtabell som returneras till *MainProgram*. Här skapas nu ett objekt av klassen *Analyzer* som kan sägas utgöra själva parsningsmotorn. XML-filen som innehåller reglerna läses in av programmet och sparas i form av Java-objekt redan i konstruktorn till *Analyzer*. Konstruktorn får även tillgång till hashtabellen som innehåller lexikonet. *MainProgram* öppnar därefter ett inmatningsfönster (se beskrivning i avsnitt 7.1 på sidan 40) och skickar med *Analyzer*-objektet som argument.

Analyzers viktigaste metod är *analyze()* som tar texten som ska parsas som inparameter och returnerar en annoterad version av texten. Inströngen lagras i en tvådimenstionell array för att kunna skilja meningarna åt. Varje ord letas upp i lexikonet som finns sparat i form av objekt i en hashtabell, med all information som behövs för respektive ordklass. Det finns olika sätt att hantera okända ord beroende på hur pass tillförlitlig algoritm man har för att gissa dessa ords PoS-annotering. Vi har inte någon sådan algoritm över huvud taget och hoppar helt enkelt över dem. Granska använder statistisk morfologisk analys av ordens ändelser och har därigenom uppnått goda resultat [4].

En nästlad loop används; den yttre loopar igenom alla meningarna och den inre alla tokens i varje mening. På så sätt gås texten igenom ord för ord. För varje ord appliceras den första regeln (som är hårdkodad som “rule1” i klassen *TextProcRule*). Så länge det finns en nästa regel specificerad fortsätter applicerandet av regler. Klassen *TextProcRule* används för detta och den söker först efter det som angivits i parsningsregelns *search* genom att anropa klassen *Search* som i sin tur anropar exempelvis *Inflection*. *Search* returnerar en boolsk variabel till *TextProcRule* och de åtgärder som specificerats i parsningsregelns *action*-del vidtas. Detta genom att olika metoder i klassen *Action* anropas, beroende på om *Search* returnerade sant eller falskt. Lämpliga räknare räknas upp och vilken som ska bli nästa regel anges.

Strängen som returneras från *analyze()* är annoterad med de taggar som specificeras i regelfilen. Den regelfil som används i den gamla versionen har en regeluppsättning för varje pronomen på det sätt som beskrevs i avsnitt 2.2 på sidan 17. Den returnerade strängen visas i ett utmatningsfönster. Någon sluttestning av den gamla versionen av Direkt Profil har inte gjorts, främst på grund av problem som fortfarande finns kvar i den versionen men som rättats till i den nya. Annoteringen verkar dock, utifrån vad vi sett i utvecklingsarbetet, fungera på ett tillfredsställande sätt.

Räknarna sparas internt i motorn i en *Map* och kan itereras igenom på det sätt som görs i den nya motorn. Innan denna funktionalitet hunnit implementeras hade vi dock beslutat att

istället gå vidare med en helt ny version av motorn, varför den gamla versionen fortfarande har räknarnas namn från XML-filen hårdkodade på det ställe i motorn där de skrivs ut. Precis som för annoteringen gäller för räknarna att de inte sluttestats, men vi har inte konstaterat några allvarliga felaktigheter under provkörningarna i utvecklingsarbetet. Det skall dock nämnas att räknarnas värden inte nollställs mellan varje parsning. I den gamla versionen av Direkt Profil får användaren alltså starta om motorn mellan varje parsning för att undvika att få räknarnas ackumulerade värden sedan programstart.

Det skall även påpekas här att den gamla versionen har den stora nackdelen att den inte klarar av franska tecken. Vad detta beror på har inte utretts, eftersom vi beslutade att lämna den gamla versionen till förmån för den helt nya avknoppningen. Detta är mycket viktigt för eventuella framtida användare att vara medvetna om; det begränsar ju starkt möjligheterna att använda den gamla versionen i mer realistiska sammanhang på autentiska texter. Franska tecken klipps nämligen bort. Om ordet *être* skrivs in och skall parsas klipps det ner till *tre* och hittas naturligtvis inte i lexikonet. Detsamma gäller *é*, varför *passé composé*-konstruktioner testats med particip som inte innehåller *é*. (Många franska particip slutar med *é*, exempelvis *commencé* och *mangé*.) Inte heller detta problem ansågs viktigt nog att lägga tid på att lösa, när vi väl tagit beslutet att överge den gamla motorn.

6.2 Den nya motorn

Den nya motorn är en vidareutbyggnad av den äldre. Det som är tillagt är anpassningar dels till det nya annoteringsformatet (som beskrevs i kapitel 3 på sidan 19) och dels till den nya regelformalismen (som beskrevs i kapitel 4 på sidan 22, se även bilaga K). De olika utfilerna skapas av en *controller*-klass med samma namn som utfilen den är ansvarig för, till exempel *PosController, SpanController, ResultController* och så vidare. Dessa klasser är nya för denna version av systemet.

Inläsning av lexikonet har inte förändrats mellan de två versionerna; det lagras fortfarande i en hashtabell. Precis som i den gamla motorn är det också *Analyzer* som samordnar analysen av texten. I metoden *analyze()* konverteras alla bokstäver i inströngen till gemener, strängen tokeniseras och varje token tilldelas ett unikt id-nummer som sparas internt. Detta skiljer sig från den gamla motorn där ju uppdelningen av tokena i meningar bibehålls i en tvådimensionell array. Därefter skapar *analyze()* ett nytt *FlowManager*-objekt som får tillgång till *TextControllers* tokens, reglerna i *RuleManager* samt några andra *controller*-klasser. *FlowManagers* metod *flowDocuments()* anropas och det är här som själva parsningsarbetet sker. Även detta är en en nyhet jämfört med den gamla motorn. Metoden består av två stora loopar som är nästlade i varandra.

Den yttre av dessa loopar igenom alla tokens i följd. De finns ju lagrade i *TextController*, men de är inte uppdelade i meningar som de var i den gamla motorn, utan parsas bara i en enda följd. Det är tänkt att satsavgränsningen ska göras i förprocessen (se kapitel 8 på sidan 45), men den har ännu inte implementerats fullt ut. Detta har lett till vissa felaktiga parsningar; exemplet *Ils prennent un café. Elle aussi mais moi, je prends un croissant.* skulle alltså, med motorns nuvarande implementation, fortfarande parsas på samma felaktiga sätt även om kommatecken angavs som satsavgränsare. Vi har föreslagit en annan lösning för att komma runt det problemet, nämligen att utöka beslutsträdet (se avsnitt 10.1 på sidan 51). Det kan dock inte helt avhjälpa att motorn underlåter att göra satsavgränsningar. Med motorns nuvarande implementation skulle nämligen de två meningarna *Je á Paris. Nous aussi.* parsas som en enda sats utan verb, trots att det egentligen är två satser utan verb.

Den inre av looparna i *flowDocuments()* går igenom de tokens som tillhör det aktuella fönst-

ret. Loopen är ganska omfattande och är uppdelad i ett antal olika fall i if-else-satser. De olika fallen består av kombinationer av olika kriterier, varav de viktigaste är:

1. Om regeln innehåller en *accord*-tagg, det vill säga om regeln är tillbakablickande.
2. Om regelns *search*-del matchade eller ej, det vill säga om *found* eller *notfound* ska appliceras.
3. Om regeln innehåller en *recursive*-tagg, det vill säga om den ska appliceras på samma token som förra regeln.
4. Om det finns någon nästa regel angiven.
5. Om man nått slutet av fönstret i kombination med att nästa regel inte är rekursiv, det vill säga att det är dags att gå ur den inre loopen.

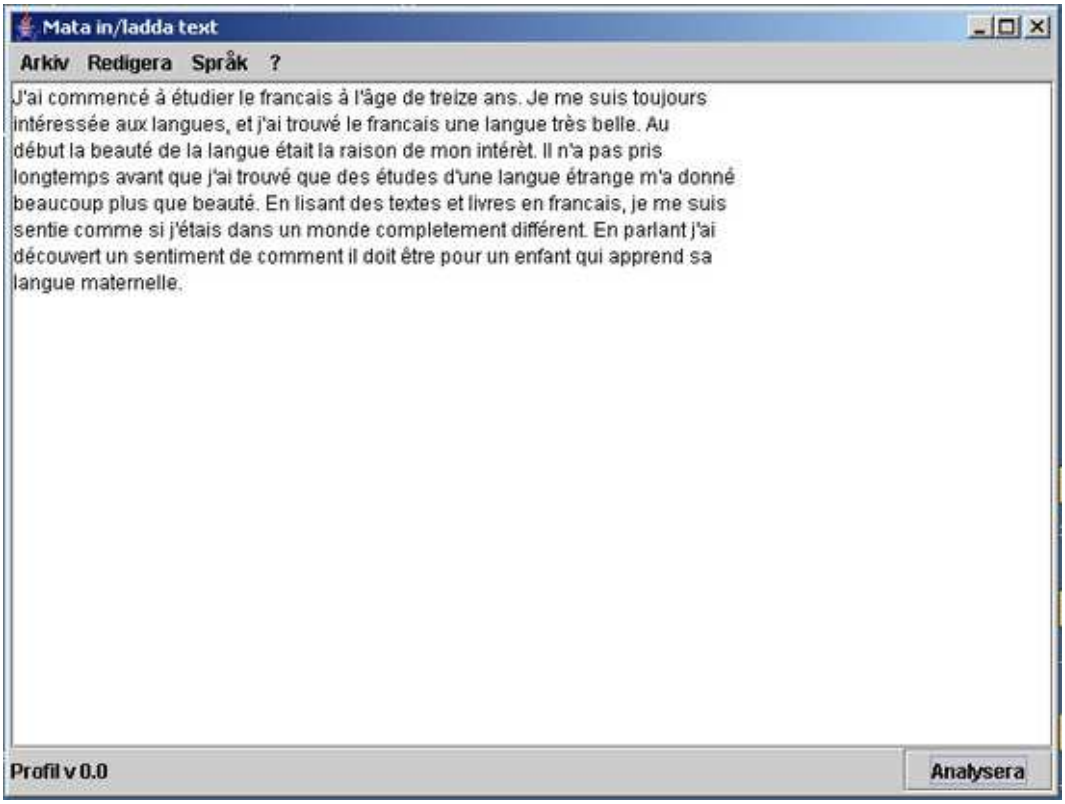
När *flowDocuments()* väl parsat alla tokens returneras kontrollen till *analyze()* som ger *ResultController* tillgång till räknarna från *FlowManager*. Räknarna lagras internt i *FlowManager* i en *Map* och denna returnerar nu en instans av *Set* med räknarnas nyckelvärden och dessa itereras igenom med en instans av *Iterator*. I *ResultController* skapas nu *counter*-elementen som ska skrivas ut i XML-format. De andra dokumenten är ju redan skapade i sina respektive *controller*-klasser. I en propertiesfil har användaren angivit vilka utfiler som ska skapas och dessa skrivs nu till disk och därefter slås de ihop till *merged.xml* som också skrivs till disk.

7 Användargränssnitten

Rapporten har hittils fokuserat på Direkt Profils funktionalitet, dess inre struktur och uppbyggnad. I detta kapitel skiftas perspektiv och systemet beskrivs sådant det ter sig för användaren. Speciellt beskrivs och diskuteras användbarheten av de två versionerna av Direkt Profil och för- och nackdelar med de två lösningarna vägs mot varandra.

7.1 Standalone-versionen

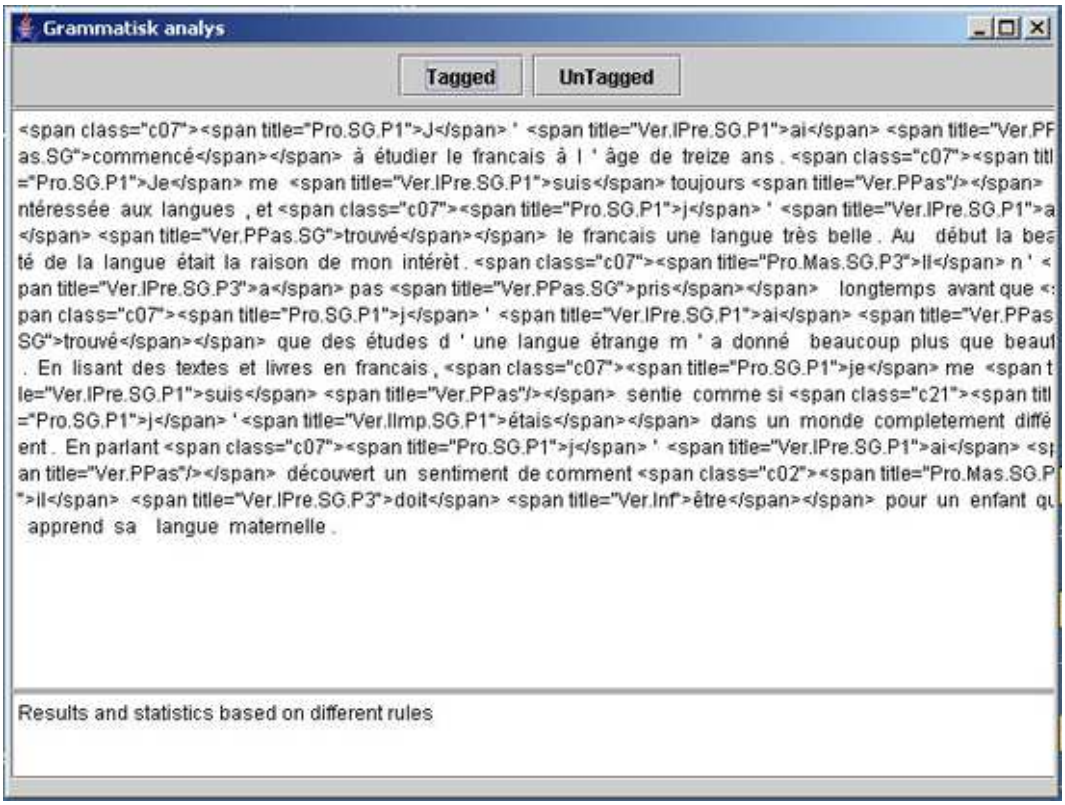
Användargränssnittet ser likadant ut i standalone-versionen i den nya versionen av Direkt Profil som i den gamla. Huvudklassen öppnar ett inmatningsfönster där användaren kan mata in sin text. Detta kan göras antingen genom att man öppnar en textfil eller genom att man skriver eller klistrar in texten direkt i fönstret. Till höger under fönstret finns en knapp med beteckningen *Analysera*. När användaren klickar på den startar motorns parsning av den inmatade texten. Den resulterande annoterade texten visas i ett nyöppnat utmatningsfönster så snart parsningen är fullbordad. I den nya versionen av Direkt Profil skrivs dessutom resultatfilerna, däribland *merged.xml*, till disk. Dessa kan sedan öppnas i exempelvis en webbläsare för att studeras närmare.



Figur 3: Inmatningsfönstret i standalone-versionen av Direkt Profil

Utmatningsfönstret blir snabbt komplext och svåröverskådligt när man har lite längre texter än bara några få ord. Även *merged.xml* blir svårhanterlig vid längre texter, eftersom man inte direkt ser de olika strukturerna upptaggade i texten. För att kunna utföra meningsfull och effektiv testning fanns alltså ett behov av ett enklare sätt att visualisera taggarna. Vi diskuterade olika

lösningar för detta, som alla hade olika för- och nackdelar ur MDI[46]-synpunkt. Vi fastnade för lösningen att skapa ett gränssnitt där taggarna visades som olika bakgrundsfärger bakom texten. Denna typ av gränssnitt är implementerat i webbversionen av Direkt Profil. De olika sätten att presentera spann- och räknar-information kan jämföras i bilagorna B till och med I å ena sidan och de två exempeltexterna i bilagorna T och U å andra sidan.



Figur 4: Utmatningsfönstret i standalone-versionen av Direkt Profil

7.2 Webbversionen

Det föreslås i KTHs annoteringsförslag [3] att verktyg ska användas för att presentera annoteringsinformationen på en läsvänligare sätt. Det är det vi gjort i och med webbversionen av Direkt Profil[47]. Användaren loggar in genom att på sedvanligt sätt ange sitt användarnamn och lösenord. Skälet till att vi valt att ha ett inloggningsförfarande i Direkt Profil är att man ska kunna ställa in personliga inställningar. Själva användargränssnittet består av fem flikar som, om man klickar på dem, visar olika gränssnitt.

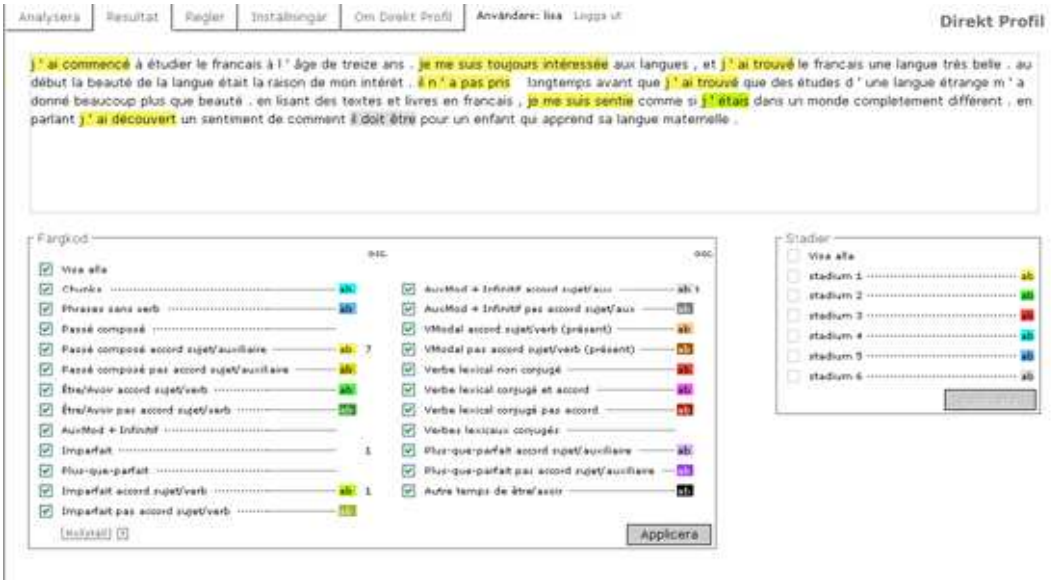
Fliken längst till vänster heter *Analysera* och liknar analysfönstret i standalone-versionen av Direkt Profil. Man kan antingen skriva in texten direkt eller kopiera in den. I framtiden är det även meningen att man ska kunna öppna en textfil direkt, men det är ännu inte implementerat, varför knapparna *Choose* och *Analysera fil* är inaktiva (gråmarkerade). Knappen *Nollställ* rensar inmatningsfönstret och knappen *Analysera text* sätter igång själva parsningen. Vi har satt en

[46] Akronym för Männikska-Dator-Interaktion.

[47] En preliminär version av Direkt Profil finns tillgänglig via www.rom.lu.se:8080/profil/

Figur 5: Inmatningsfönstret (fliken *Analysera*) i webbversionen av Direkt Profil

begränsning på 6282 tecken i webbgränssnittet för att inte det ska bli så långa svarstider. När parsningen är klar öppnas automatiskt nästa flik, *Resultat*, där den annoterade texten visas.



Figur 6: Utmatningsfönstret (fliken *Resultat*)i webbversionen av Direkt Profil

I fliken *Resultat* märks den stora fördelen med webbgränssnittet framför det gamla sättet att presentera spann och deras respektive räknare på. Fönstret är uppdelat i tre fält; överst en textruta och därunder två rutor vid namn *Färgkod* respektive *Stadier*. I textrutan visas texten, konverterad till små bokstäver rakt igenom och med varje token visat med ett blanksteg mellan, varför även alla kommateringstecken, till exempel punkter, föregås av ett blanksteg. Varje annoterad struktur är markerad med den bakgrundsfärg som motsvarande räknare har. Då musmarkören dras över de markerade orden kommer en liten ruta upp med den ordklass- och särdrags-taggning som getts det aktuella ordet. Denna funktion har visat sig användbar i de fall Direkt Profil inte taggat som förväntat, till exempel på grund av ambiguitetsproblem som uppstår till följd av lexikonets

uppbyggnad (se avsnitt 5.3 på sidan 34).

I rutan färgkod visas räknarna i två kolumner; till vänster i varje kolumn räknarens namn och till höger dess färg och antal förekomster i texten. Bokstäverna "ab" som står i rutorna visar vilken färgning själva texten får; med de mörkare bakgrundsfärgerna är texten vit. Vi har valt att ge räknarna de franska namn de initialt hade i beslutsträdet. Förklaringar till räknarnas namn finns i bilaga S.

Många av fenomenen som Direkt Profil annoterar uppträder i par där den ena typen av struktur är korrekt och den andra felaktig. I dessa fall färgas de korrekta med en ljusare och de felaktiga med en mörkare ton av samma färg. Förekomster av *être/avoir* med kongruens färgas mellangrönt, medan *être/avoir* utan kongruens färgas mörkgrönt. Satser utan verb färgas blå och har ingen "tvillingräknare". Det har inte heller andra tempus/modus för *être/avoir*. De färgas svarta. Dessutom finns fem räknare som alla är vita. Även "ab" är vitt här eftersom de räknarna aldrig visualiseras i texten. De är ju summan av andra, mer specifika räknare (för detaljer, se avsnitt 4.7 på sidan 29) och utgör alltså aldrig större spann än de mer specifika, varför de aldrig kommer att synas i texten.

Rutan *Stadier* är inaktiv (knappen *Applicera* är gråfärgad) eftersom detta ännu inte implementerats. Vi valde att ändå ha med denna ruta i gränssnittet dels för att visa på vår långsiktiga idé med Direkt Profil och dels för att framtida utvecklare av Direkt Profil ska slippa strukturera om användargränssnittet.

Under fliken *Regler* visas först räknarna med respektive färgning och sedan parsningsreglerna för användaren. I regeldelen är både räknarnas namn och *nextrule* klickbara och leder till räknare respektive regel. Användaren kan inte editera reglerna här, utan bara titta på dem. Under fliken *Inställningar* kan användaren ange sina personliga inställningar vad gäller exempelvis storlek på inmatningsfönstret eller språk i användargränssnittet. Fliken *Om Direkt Profil* är tom än så länge. Det är meningen att sedvanlig så kallad about-information ska finnas här.

Den är så kallade CSS[48]-filer som översätter XML-filerna till webbfomat. De är hårdkodade efter regelformalismen. Man kan dock lägga till regler och räknare utan att behöva ändra i CSS-filerna.

7.3 Webbgränssnittet ur användbarhetssynpunkt

Frågan om ett systems användbarhet är ett helt forskningsområde i sig. I utvecklandet av Direkt Profils användargränssnitt har vi endast använt oss av de grundläggande kunskaper vi har i ämnet MDI. För att kunna optimera gränssnitten krävs att empiriska användbarhetsstudier genomförs, något som inte varit aktuellt i detta arbete. Detta avsnitt syftar endast till att skissera huvuddragen i de resonemang som föregått de designval vi gjort i utformandet av webbgränssnittet.

Interaktionen med Direkt Profil är intuitivt med både webbgränssnittet och standaloneversionens dito. Vi har utformat flikarna så att de ligger i logisk ordning, först *Analys* och sedan *Resultat*. Det är ju främst dessa flikar som används vid textparsning med Direkt Profil. Sedan ligger flikarna i sjunkande ordning vad gäller användarfrekvens.

Huruvida färger ska vara betydelsebärande vid utformningen av användargränssnitt är alltid en svår fråga. Den stora fördelen är naturligtvis att det är ett enkelt och tydligt sätt att visa information (vilket skillnaden i användarvänlighet mellan standalone-versionen och webbversionen tydligt visar). Den stora nackdelen är att en sådan lösning ofrånkomligen utesluter en viss del av användarna. Inte bara de som lider av en eller annan form av färgblindhet, utan även de som på grund av ålder eller av andra orsaker inte kan tillgodogöra sig nyansskillnaderna i närliggande

[48] Cascading Style Sheet

färgtoner. Även för en person med mycket god syn har det visat sig svårt att skilja mellan de två färger som markerar *VModal pas accord sujet/verbe* och *Verbe lexical conjugué pas accord* i webbgränssnittet (se bilaga S). Färgvalen som föreligger i den aktuella versionen av Direkt Profil kan naturligtvis omarbetas och förbättras ur användbarhetssynpunkt.

Ett annat problem med att använda färgskillnader som markör är att de är ändliga till antalet, åtminstone de som på ett tydligt sätt kan skiljas från varandra. Dessa problem slipper man med standalone-versionens presentation av annoteringen.

Ett problem som inte är unikt för Direkt Profil, utan som alltid finns på webben är att gränssnitt ser olika ut i olika webbläsare. Ett exempel är när man klickar på räknarnas namn i fliken *Regler*. I vissa webbläsare visas inte räknaren överst i fönstret, utan själva räknarens definition.

Ett annat problem med färger på webben kan vara att alla datorer inte kan visa alla färger. Detta problem har blivit mindre och mindre i takt med den tekniska utvecklingen och är knappast att betrakta som särskilt viktigt.

8 För- och efterprocess

På lite längre sikt är det tänkt att Direkt Profil ska arbeta i tre steg. Hittils har systemet beskrivits som bestående av tre huvuddelar, nämligen regler, lexikon och parsningsmotor. Dessa tre delar utgör dock endast det mittre av dessa steg, den så kallade huvudprocessen. Den är den mest omfattande av de tre och dess implementation är i stora drag färdig, vilket beskrivits ovan. Trestegsförfarandet ser schematiskt ut så:

Förprocess: Söker efter och taggar upp chunks. Sparar undan information om stor bokstav. Sparar undan information om satsavgränsningar (*delimiters*). Gör rent statistiska beräkningar på orden i texten.

Huvudprocess: Söker igenom texten efter de strukturer som reglerna definierar. Dessa taggas och räknas. Det är detta steg som det här examensarbetet framförallt tar sikte på och som beskrivits i ovanstående kapitel.

Efterprocess: Gör en analys av inlärarens språknivå enligt tabellen för utvecklingsgångar utifrån de olika räknarnas värden. (Delar av denna tabell [2] visades i avsnitt 1.2 på sidan 12.)

Vi har börjat förbereda systemet för förprocessen. Framförallt är det utformandet av en lösning för att kunna implementera chunks som vi påbörjat. Nedan finns en beskrivning av resultaten från det arbetet och sedan följer våra implementeringsförslag för några av de andra uppgifterna som förprocessen ska kunna utföra.

Vad gäller efterprocessen har ingen implementation eller ens design påbörjats. Ett mål med Direkt Profil är att systemet automatiskt ska kunna göra en nivåbedömning av en text, kunna "profilera" en text. Det återstår ännu att utveckla en formalism för att koda stadieindelningen. Detta skulle visserligen kunna göras på ett något rudimentärt och provisoriskt sätt redan nu, men vi föredrar att vänta med detta tills sökverktyget har använts i forskningen för att precisera de skriftliga utvecklingsgångarna.

8.1 Chunksproblematiken

Det finns lite olika innebörder i ordet "chunk". Här ska chunks förstås i den traditionella lingvistiska betydelsen av ordet, och inte i exempelvis Steven Abneys betydelse där till exempel nominalfraser kan utgöra *noun chunks (NX)*, verbfraser kan utgöra *verb chunks (VX)* och så vidare. Det finns ganska strikta regler för vad som räknas till en viss typ av chunk enligt Abneys definition och hur långt chunken sträcker sig [1].

Chunks (i den betydelse som avses här) är den lingvistiska benämningen på fixa uttryck som lärts utantill och som används utan djupare förståelse av deras grammatiska uppbyggnad. Ett exempel på en sådan struktur är *je m'appelle* som varje nybörjare i franska vet betyder *jag heter*. Det är dock inte självklart att man känner till vad de olika orden i uttrycket betyder (eller åtminstone inte deras syntax). Inläraren använder dessa fraser, chunksen, i sin helhet. De används nästan som "trollformler", utan att inläraren kan anpassa dem i nya sammanhang. Man behärskar alltså inte automatiskt *tu t'appelles*, bara för att man kan använda *je m'appelle*. De chunks som hittils skrivits in i chunksfilen listas i bilaga Q.

Skälet till att vi vill särbehandla chunks i Direkt Profil är att de har en syntaktisk konstruktion som är för avancerad för den språkliga nivån i övrigt. Om de skulle parsas som vanligt skulle användandet av *je m'appelle* tyda på en högre nivå än vad språkinläraren kanske i själva verket uppnått. Detta eftersom det antyder en regeltillämpning som inte finns, både av verb som böjs efter sitt pronomen, av klitiska pronomen, och av korrekt ordföljd. Användning av *tu t'appelles* däremot visar troligen korrekt på denna högre nivå, eftersom det inte är en chunk man använder sedan första dagen man studerat franska.

Två olika lösningar för chunksproblematiken har diskuterats. Det ena alternativet var att helt enkelt lista chunksen i en vanlig textfil, vilket är enklast för användaren av programmet när denne vill prova att lägga till ytterligare chunks. Dessutom verkade detta tillvägagångssätt enklast och snabbast att implementera. De mönster som fanns med i den filen skulle helt enkelt kunna tas bort ur texten innan den skickades till parsningsprocessen. Detta leder dock till två följdproblem:

1. Det kan vara nödvändigt att lagra information om vilken chunk som hittats var. Denna information används när man kontrollerar chunkens omedelbara omgivning. En chunk som *je voudrais* förväntas följas av ett verb i infinitiv eller en nominalfras. Verbet i infinitiv ska förmodligen annoteras och räknas för att räknarna ska kunna ge en korrekt nivåindikation. Om chunken helt sonika tagits bort kan dess kontext inte kontrolleras.

2. Man kan inte räkna hur många av varje sorts chunk som hittats eftersom man inte har några räknare kopplade till dem. Sådana räknare skulle i så fall få skrivas in i motorn, vilket strider mot vår strävan att göra motorn så generell som möjligt.

Det andra alternativet för lösning på chunksproblematiken var att utveckla en XML-formalism för chunks, som skulle appliceras innan reglerna i en förprocess. Detta alternativ blev det naturliga valet med tanke på den utökade funktionalitet som XML tillåter jämfört med rå text. Man kan koppla en räknare till varje chunk och dessutom en identifierare som skulle kunna användas på liknande sätt som i Persona-projektet (Ball et al 1997, refererad av Nugues [19]). I det projektet byter man ut formeluttryck[49] mot identifierare. Man slipper på så sätt parsa låttexter innehållande flera ord, vilket förenklar senare analys av den omedelbara kontexten. Detta sker på ett förstadium; det handlar alltså inte om analys utan fördetektion och räkning.

Det finns även andra fördelar med valet att använda XML. Det gör att Direkt Profil konsekvent använder XML för de kodningar som är språkspecifika; både reglerna och lexikonet ska ju vara XML-kodade. Dessutom undviker man på detta sätt problem med teckenuppsättningen. Vårt förslag till chunksformalism ser ut på följande sätt:

```
<chunk size="2" identifier="chunk2">
   <token id="1" value="je" nextid="2"/>
   <token id="2" value="voudrais"/>
</chunk>
```

Taggen *chunk* har två attribut. Det första av dessa är *size* som anger antalet tokens som ingår i chunken. Man måste hålla i minnet att även skiljetecken räknas som egna tokens. Exempelvis har chunken *je m'appelle* fyra tokens, trots att den bara består av tre ord, eftersom apostrofen räknas som ett eget token. Man måste inte ange *size*. Det kan försvåra för den som skriver XML-filen att se till att *size* blir rätt. Det andra attributet *identifier* är identifieraren som infogas i texten i stället för chunken innan den parsas.

Chunk-taggen har en barntagg, nämligen *token*. Att chunken är uppdelad i sina tokens istället för att skrivas som en enda sträng underlättar förprocessandet eftersom tokeniseringen på de här stadiet redan är genomförd. *Token*-taggen har tre attribut, men inga barntaggar. Det första attributet är ett chunkunikt *id* på liknande sätt som används i annoteringsformatet (se kapitel 3 på sidan 19). Det andra attributet, *value*, anger själva tokenet och det sista attributet, *nextid*, anger vilket token som följer på detta i chunken. Attributet *nextid* underlättar genomsökningen av den tokeniserade texten genom att man direkt vet vilket som förväntas vara nästa ord för att

[49] Formeluttryck är inte riktigt chunks i vår bemärkelse, men idén med identifierare är densamma

det ska kunna vara den sökta chunken. Bearbetandet av chunks är som sagt ännu inte implementerat i Direkt Profil, utan vi har endast påbörjat utvecklingen av den funktionaliteten. Efter att utvecklingen av en chunksformalism redan påbörjats har andra saker som ansetts viktigare upptäckts.

8.2 Spara undan information om stor bokstav

I den nuvarande versionen av Direkt Profil konverteras alla ord till gemener, och skrivs även ut så i resultatfilerna. Skälet till denna konvertering är att lexikonets alla uppslagsord står med gemener och sökningen i lexikonet är skiftlägeskänslig (eng. *case sensitive*). Detta har inte lett till några problem i vår implementation, men det kan leda till tvetydigheter i fall då begynnelsebokstavens skift (eng. *case*) är betydelsebärande. Exempelvis betyder *pierre* sten på franska, medan *Pierre* är ett inte helt ovanligt egennamn. Informationsförlusten vid konverteringen till gemener måste lösas för att undvika problem i framtiden, särskilt som ett av de långsiktiga målen med Direkt Profil är att det ska kunna användas även för andra språk. Frågan måste definitivt beaktas om systemet ska används för tyska, där ju substantiv har versal begynnelsebokstav. Ordet *Laden* betyder butik och är alltså ett substantiv, medan det, om det skulle konverteras till *laden*, skulle betyda lasta och alltså vara ett verb. I sådana fall är det nödvändigt att konverteringen från gemener till versaler förfinas eller till och med förändras helt, för att inte parsningen ska bli felaktig.

Ett förslag till lösning är att konvertera alla ord till att ha versal begynnelsebokstav, både i lexikonet och texten som ska parsas. Detta skulle främst vara aktuellt för tyska där som sagt versal eller gemen begynnelsebokstav är av större vikt än i franska.

En alternativ lösning skulle kunna vara att inte konvertera orden och sedan göra sökningen i lexikonet skiftlägesokänslig.

Båda dessa lösningar förkastades dock till förmån för förslaget att lägga till ett extra attribut i *token*-taggen som anger om ordet hade versal eller gemen begynnelsebokstav innan konverteringen. Detta går mer i linje med med de tidigare designvalen att använda XML där det är möjligt. Dessutom är det mer kraftfullt eftersom den informationen kan komma att användas i fler sammanhang i systemet och det är inte så oflexibelt som de andra lösningarna. Med dem går ju informationen förlorad; *pierre* kommer inte kunna särskiljas från *Pierre*.

8.3 Ord med felaktig morfologi

När inläraren väl förstått och börjat tillämpa en regel görs detta med den regelbundenhet som även barn använder. Barnet som lär sig svenska säger *gådde* innan det säger *gick* och *skärde* innan det säger *skar*. Samma sak gör andraspråksinläraren som vet att particip av re-verb konstrueras genom *stam+u*, och som därför ibland konstruerar *prendu* ur verbet *prendre*. Detta är dock inte korrekt eftersom prendre är oregelbundet; participformen heter *pris*. Skälet till att acceptera den här typen av ord, trots att de inte är korrekta, är att de tyder på en regeltillämpning som man är ute efter att nivåbestämma i Direkt Profil. I det här fallet visar användning av *prendu* en medvetenhet om hur particip konstrueras.

Ett sätt att implementera detta kan vara att de ord med felaktig morfologi som man är intresserad av ligga i en separat lexikonfil. Om ett ord inte hittas i det ordinarie lexikonet går man till detta alternativa lexikon för att se om det finns där.

8.4 Statistiska beräkningar i texten

Även om syftet med Direkt Profil är att göra en kvalitativ analys av inlärartexter, kan även kvantitativa metoder användas som komplement. Det finns många olika fenomen att ta fasta på som är enkla både att detektera och räkna. Nedan följer exempel på tre rent statistiska uppgifter som kan vara en nog så god indikator på vilken nivå en text befinner sig på. Dessa fenomen kan ge ytterligare en dimension åt nivåbedömningen och sammantaget med den kvalitativa analysen ge en mer rättvis bild åt en text.

1. Räkna antal ord per mening. Generellt sett tyder korta meningar på lägre språknivå än långa.

2. Räkna antal olika lemman i texten, vilket ger en fingervisning om författarens ordförråd; ju fler lemman, desto högre språknivå.

3. Räkna antal okända ord, vilket kan tyda på många felstavningar. Detta i sig behöver inte nödvändigtvis tyda på låg språknivå, se diskussion i avsnitt 10.1 på sidan 51. Hög frekvens okända ord kan också tyda på att man "skarvat i" med ord från andra språk som man förfranskat. Exempel från våra exempeltexter är *try* och *during*. I sådana fall tyder det sannolikt på lägre språknivå. Detta att räkna okända ord måste ske mot det stora lexikonet och kan inte göras med ett lexikon som bara innehåller pronomen och verb (som diskuterats i avsnitt 5.2 på sidan 32).

9 Resultat

Direkt Profil befinner sig fortfarande i ett inledande skede av utvecklingsarbetet. Det föreligger i dagsläget (augusti 2004) en körbar prototyp på vilken vi utfört preliminära tester. Någon mer omfattande, kvantitativ testning av systemet har ännu inte ansetts meningsfull. Denna sluttestning är snarare kvalitativ och består av tolv autentiska inlärartexter. Texterna är av varierande längd och befinner sig på olika språkliga nivåer. Testningen sker med den senaste versionen av regelfil (*rules14.xml*) som implementerar hela det träd som presenterades i avsnitt 4.1 på sidan 22.

Det lexikon som används är det stora råtextlexikonet, dico. Ur det har vi tagit bort vissa pronomen som under systemutvecklingen visade sig orsaka många falska taggningar av mening utan verb och andra fel. Dessa listas i bilaga L.

En annan åtgärd som vidtagits för att ge en så rättvisande bild som möjligt av Direkt Profils detekteringsförmåga är att vi rättat stavfel på verb och pronomen. Testningen har skett med hjälp av webbversionen av Direkt Profil, och den resulterande parsningen visas i bilagorna R.1 till och med R.12. Som brukligt är sker testningen med texter som är helt nya för systemet och som alltså inte använts i utvecklingsarbetet.

9.1 Precision- och recallberäkningar

Precision och *recall* är två mått som brukar användas i datalingvistiska sammanhang för att ge en uppfattning om hur väl ett system lyckas hitta de strukturer man letar efter.

$$Precision = \frac{Antal\ korrekt\ detekterade\ strukturer}{Totalt\ antal\ detekterade\ strukturer}$$

Precision visar hur många av de strukturer systemet hittar som är korrekt detekterade. Övriga strukturer är alltså inkorrekt taggade. Detta mått visar, som namnet antyder, med hur god precision systemet arbetar.

$$Recall = \frac{Antal\ detekterade\ strukturer}{Totalt\ antal\ strukturer}$$

Recall visar hur stor del av de strukturer som fanns i texten som faktiskt hittades av systemet. Detta mått ger alltså en uppfattning om hur väl programmet lyckas hitta strukturer överhuvud taget i texten.

Det ska påpekas att resultaten från dessa tester endast är en fingervisning om Direkt Profils förmåga. Testtexterna är kommenterade i bilagorna, dels med vilka siffror som ligger till grund för *precision*- och *recall*-värdena, dels med vilka fel som orsakat att värdena inte är 100% (i förekommande fall) och dels med kommentarer om andra egenheter. Det kan röra sig om detekterade strukturer som i sig inte är felaktigt uppmärkta, givet de regler och det lexikon som Direkt Profils nuvarande version arbetar med, men som en mänsklig parsare skulle taggat annorlunda.

	Precision	Recall
Hans	83%	100%
Roland	100%	100%
Lillie	50%	100%
Lena	80%	100%
Thea	100%	100%
Amy	100%	100%
Nellie	100%	100%
Ella	91%	100%
Christine	100%	100%
Nicole	95%	100%
Ingmar	100%	100%
Inga	100%	100%
Samtliga texter	96%	100%

Att *recall*-värdena är 100% rakt över är inte så förvånande med tanke på att en struktur alltid taggas om den inleds med ett pronomen. Det innebär att så länge den funktionaliteten fungerar som den ska, kommer alla strukturer att detekteras som ska detekteras. Det säger dock ingenting om huruvida de är korrekt markerade eller ej. Det gör däremot *precision*-måttet, varför det egentligen är mer intressant att titta på i dessa tester. *Precision*-värdena är överlag relativt höga och det kan konstateras att den enskilt största felkällan som sänker värdena är problemet med ordningen i lexikonet (som diskuterades i avsnitt 5.3 på sidan 34). Det sammantagna *precision* för alla strukturerna i texterna ligger på 96%, vilket är ganska högt.

9.2 Prestanda

Prestanda har inte varit en prioritet i arbetet med Direkt Profil. Sådana hänsyn har åsidosatts till förmån för annat som ansetts viktigare, främst att se till att systemet annoterar och räknar så rätt som möjligt. Vi ansåg att engångsinläsningen av lexikonet i princip får ta hur lång tid som helst och att parsningstiden anses acceptabel om den ligger under 20 sekunder för 10 rader text (ca 250 tokens).

På en dator med Pentium4-processor (2,8 Mhz) och 1,5 gigabyte RAM tar det 16 sekunder att läsa in det stora (icke XML-formaterade) lexikonet. När det väl är gjort tar det cirka 35 sekunder att parsa en text om 2 A4-sidor (cirka 1450 tokens).

Samtidig analys fungerar inte alltid i webbversionen. Flera användare kan dock vara inloggade samtidigt, men inte skicka förfrågningar samtidigt. I sådana fall händer det ibland (men inte alltid) att det inte kommer ut någon färgning och att räknarnas värden blir felaktiga (för höga). Vi har även testat att en användare kör en parsning som tar lång tid och under tiden skickar en annan användare en kortare förfrågan. Detta fungerar sånär som på att båda får den kortare texten i analysfönstret när de återvänder till den fliken.

10 Avslutande diskussion

Syftet med detta examensarbete var att utveckla och förbättra den befintliga versionen av Direkt Profil. Det skulle göras genom att det beslutsträd för verb som finns framtaget inom ramen för forskningsprojektet helt och hållet implementerades. Det konkreta målet var att systemet ska kunna identifiera, klassificera och räkna mönster automatiskt och utifrån den resulterande utvärderingen kunna nivåbestämma den enskilda inläraren, samt att systemet skulle vara så stabilt att det går att använda i den lingvistiska forskningen kring franska utvecklingsgångar i skrift.

Vi har med Direkt Profil skapat ett system som kan identifiera och klassificera vissa grammatiska mönster, nämligen verbformer i samband med pronominella subjekt i en text på franska och även annotera dessa mönster. Systemet räknar dessutom de funna mönstren i olika kategorier. Vi har därmed lyckats ta fram ett program som är praktiskt användbart i språkforskningen kring utvecklingsgångar i skriven franska. Systemet är också så pass stabilt att det nu kan börja testas i större omfattning i mer autentiska sammanhang.

Ett långsiktigt mål med Direkt Profil som ännu inte uppnåtts är att systemet automatiskt ska kunna nivåbestämma en parsad text utifrån de olika räknarnas värden. Detta måste implementeras innan systemet kan användas för sitt praktiska syfte, nämligen att hjälpa studerande med en fingervisning om deras språkliga nivå. För att göra detta krävs att efterprocessen implementeras. Detta har dock inte varit det primära målet i utvecklandet med systemet; huvudsyftet var ju att skapa ett verktyg som kunde användas i språkinlärningsforskningen.

Under arbetets gång har vi funnit många aspekter av systemet som bör förbättras. Några av dem är av högre prioritet än andra och nedan redogörs först för dessa och sedan nämns några förbättringar som tar längre tid eller är mer designkrävande.

10.1 Kortsiktiga vidareutvecklingar av Direkt Profil

Nästa steg som Direkt Profil står inför i och med detta examensarbetes slutförande är implementering av särskiljande mellan olika typer av pronomen. Detta kan göras genom att ytterligare särdragsinformation läggs till i lexikonet om pronomina i kombination med att beslutsträdet utökas så att det specifikt söker efter subjektspronomen.

I framtida versioner av det nuvarande beslutsträdet bör infogas sökning efter nytt pronomen i samband med att man letar efter ett verb. Med det utseende trädet har nu kommer (den felaktiga) satsen *nous savons que elles savoir* att parsas som modalt hjälpverb med efterföljande infinitiv (nod 5g). Denna parsning stämmer inte, eftersom *savoir* inte hör till *nous savons* utan till *elles*. Att det förekommit ett nytt pronomen efter *nous savons* upptäcks inte eftersom man söker efter verb i infinitiv och inte pronomen. (Den korrekta parsningen är modalt verb med kongruens mellan subjekt och verb (nod 5d) samt lexikalt, icke personböjt verb (nod 6b)). Kontroll om nytt pronomen förekommit behövs i samband med nod 3, nod 4a och nod 5a.

En sådan implementation är dock inte heller problemfri. Ett exempel är *nous voulons vous accompagner* där *vous* inte skall tolkas som nytt subjektspronomen, utan är objektspronomen i satsen. Med trädets nuvarande utseende parsas meningen som modalt hjälpverb med efterföljande infinitiv (nod 5g), vilket är det önskade. Med den diskuterade utökningen av trädet skulle meningen parsas som två skilda strukturer, nämligen modalt hjälpverb (nod 5d) och icke personböjt lexikalt verb (nod 6b). Detta kan lösas genom att man särskiljer subjekts- och objektspronomen i lexikonet, något som snart kommer att implementeras.

Stavfel är något som en mänsklig nivåbedömare av texten skulle förstå och överse, eller kanske inte ens lägga märke till vid genomläsning av texten. För programmet däremot gör förekomst av *avior* istället för *avoir* att de grammatiska reglerna inte fungerar eftersom man inte

hittar ordet i lexikonet. Visserligen kan stavfel också påverka vid helhetsbedömningen av en texts språknivå, men det är inte stavfel Direkt Profil ska kontrollera. Dessutom är det lätt hänt att slinta på tangentbordet och råka skriva fel på en bokstav i ett ord.

På svenska tangentbord saknas vissa av de franska specialtecknen, exempelvis *ê* eller *ç*. Denna opraktiska egenskap kan göra att man frestas skriva *etre* istället för *être* och *francaise* istället för *française*, vilket ger samma effekt som om orden varit felstavade "på riktigt". Den lösning som i nuläget föreslås för att undkomma detta problem är att man i samma fil som orden med felaktig morfologi (eller i en separat fil) lägger till vanliga felstavningar på ord och PoS-taggar dem som det rättstavade ordet. På så sätt kan Direkt Profil hitta en PoS-taggning för ordet, applicera reglerna på strukturen och ge en mer rättvisande bild åt textens språknivå.

Detta löser dock inte alla problem med att vissa bokstäver saknas på svenska tangentbord. Även om det finns möjlighet att skriva *é* på ett svenskt tangentbord (bokstaven finns ju även i svenskan) kvävs det ofta två tangenttryckningar för att åstadkomma *é*. Detta ökar risken för att bokstaven skrivs in som *e*, inte bara av lathet; orsaken kan även vara okunskap om vilka knapptryckningar som frambringar *é*. Här öppnar sig alltså nya datalingvistiska frågor om huruvida ett inskrivet *commence* istället för *commencé* är ett utslag av okunskap om hur tangentbordet eller participformen fungerar. Om det skulle visa sig vara den första förklaringen, alltså bristande förståelse för hur *é* konstrueras med ett svenskt tangentbord, hur ska vi i så fall kunna skilja detta från det andra fallet? *Commence* är ju en korrekt verbform på franska, nämligen presensformen av det lexikala verbet *commencer* böjt i första eller tredje person singularis. Satsen *j'ai commence* kommer därför parsas som nod 4d (*être/avoir* med kongruens, men utan efterföljande particip), när det i själva verket skulle motsvara nod 4g (*être/avoir* med kongruens och efterföljande particip). Eftersom participformerna mycket ofta innehåller *é* och *passé composé* är en vanlig konstruktion, särskilt på vissa inlärarnivåer, kan detta problem ge ganska stora utslag i räknarna och därmed nivåbedömningen.

Ett sätt att lösa problemet med att användaren inte vet hur hon ska åstadkomma vissa specialtecken kan lösas genom att knappar läggs till i användargränssnittet som man kan klicka på för att infoga franska tecken i inmatningsfönstret. Dock är detta ett ganska icke-spontant sätt att skriva och hjälper inte i de fall texten klistras in i inmatningsfönstret eller öppnas direkt i en textfil.

Det verkar relativt lätt att åtgärda problemet med att alla ord urskiljningslöst konverteras till gemener och står därför högt på prioritetslistan för framtida vidare utvecklingar för Direkt Profil.

Bland de verb som är definierade som modala hjälpverb står inte *aller* upptaget. Detta tillägg i kombination med adekvat utökning av beslutsträdet skulle göra att även *future proche*-konstruktioner skulle kunna detekteras av Direkt Profil. Även detta skulle gå ganska enkelt att genomföra eftersom lite eller ingen ny design behövs; implementationen skulle kunna ske analogt med den tidigare sökningen efter infinitiv i beslutsträdets gren 5.

10.2 Långsiktiga vidareutvecklingar av Direkt Profil

Kostadinov/Thulin föreslog i sin rapport att sökkriterier sammanlänkas med de logiska operatorerna AND och OR. Om regelformalismen utökades på det föreslagna sättet skulle det kunna reducera några av de extraregler som utgör en av de skillnader som finns mellan det ursprungliga beslutsträdet och det regelanpassade dito (se avsnitt 4.1 på sidan 22).

Arbetet med chunks-formalismen är påbörjad och kan utvecklas på medelfristig sikt i forskningsprojektet. Särskiljandet mellan olika typer av pronomen har hittills ansetts högprioriterat och andra utvecklingsriktningar, till exempel chunks-utvecklingen har fått stå tillbaka till för-

mån för detta. Dock bör det arbetet kunna återupptas och slutföras inom överskådlig tid när problematiken kring pronomina väl är löst.

Vi har diskuterat om det är lämpligt att användaren av Direkt Profil själv ska kunna redigera reglerna. Detta skulle i så fall kunna göras genom ett grafiskt gränssnitt där man kan välja olika alternativ ur skrollistor. Denna möjlighet föreslogs redan i Kostadinov/Thulins rapport [17]. Det är ju naturligtvis nödvändigt att språkforskaren i sitt arbete med Direkt Profil ska kunna ställa in reglerna efter sina aktuella behov och även kunna prova och jämföra olika alternativ. Åtminstone fönsterstorleken måste kunna varieras (se diskussionen i avsnitt 4.5 på sidan 27).

Däremot är det inte lika säkert att den enskilde språkinläraren eller ens dennes lärare ska kunna göra den typen av ändringar. I så fall får man göra en avvägning om det är möjligt och i så fall rimligt att lingvisterna som arbetar med Direkt Profil förväntas gå in direkt i XML-filerna och göra ändringarna där. Om svaret på den frågan befinns vara ja, är utvecklandet av ett grafiskt gränssnitt för att redigera reglerna onödigt. I nuläget är webbgränssnittet uformat så att man i framtiden kan lägga till funktionalitet med skrollistor under fliken Regler.

Inte alla verb i lexikonet har information om vilket genus de är böjda efter. Det behövs ju inte i de flesta fall, eftersom det inte syns på verbböjningen om pronomenet är femininum eller maskulinum. I de fall det förekommer, exempelvis i vissa participformer där femininformen får ett extra avslutande *e*, finns den informationen i lexikonet. Dock vore det mer generellt om alla verb kodades med information om genus. Då skulle inte formalismen behöva ändras vid överföring till andra språk. Nackdelen med detta är naturligtvis att ju mer information man har, desto större är risken att man gör fel; det kräver mer av den som skriver XML-reglerna. Detta är en utbyggnad som måste diskuteras närmare innan det kan bli aktuellt med en implementering.

En del ord har hög felstavningsfrekvens. Orsakerna därtill kan variera (se avsnitt 10.1 på sidan 51), men i alla händelser vore det önskvärt om Direkt Profil kunde har överseende med stavfel; det skulle ju en mänsklig läsare haft. *Stava* är ett KTH-projekt som implementerar olika rättstavningsalgoritmer applicerade på svenska. Stava använder bland annat ordfrekvensinformation för att ranka de föreslagna korrigeringarna [15]. På ett senare stadium i utvecklandet av Direkt Profil kan man implementera approximativ matchning, det vill säga att ord som är felstavade förstås av programmet och därmed också kan parsas på rätt sätt. Det skulle i så fall ersätta den kortsiktiga lösningen att använda en listning med felstavade ord som accepteras av Direkt Profil som föreslogs ovan i avsnitt 10.1 på sidan 51.

Att göra tillägget med ord med felaktig morfologi som beskrevs i avsnitt 8.3 på sidan 47 bör vara relativt enkelt. Det har dock inte ansetts högprioriterat i Direkt Profil eftersom det inte förväntas ge så stora förbättringar i resultaten.

En begränsning som Direkt Profil har är att spann inte kan överlappa varandra. Det innebär att de två taggningarna av meningen *Ceci n'est pas une pipe* inte kan presenteras i samma frasinformationsfil. (De två taggningarna skulle bestå i token 0-3 *être/avoir* med kongruens och token 1-4 negation. Denna senare taggning är dock inte implementerad.) Detta har inte medfört något problem för oss i våra tester, eftersom endast ett beslutsträd är implementerat än så länge. En möjlig lösning kan vara att man helt enkelt skriver ut resultatet från olika träd i olika frasinformationsfiler. Hur olika beslutsträd ska hanteras är för övrigt en fråga som vi lämnar öppen.

Den viktigaste förändringen inför nästa version av Direkt Profil är att kunna söka även på icke-pronominella subjekt, alltså nominalfraser som *la femme, deux petit chiens* och så vidare. Detta kräver en grundlig diskussion om hierarkin i systemet och en mycket långsiktig förändring i Direkt Profil är att man ändrar strategi för beslutsträdet. Denna förändring skulle kunna innebära att man söker efter verb i förstone och sen, utifrån detekterade verb, söker deras tillhörande pronomen.

11 Referenser

[1] Abney, Steven, *Chunk Stylebook* (1996), opublicerat manuskript från webbplatsen `http://www.vinartus.net/spa/publications.html`, den 12 maj 2004

[2] Bartning/Schlyter (i tryck): *Itinéraires acquisitionnels et stades de développement en francais L2* (2004) Utkommer i "Journal of French Language Studies" i december 2004.

[3] Bigert/Kann/Knutsson/Sjöberg, *Korpusformat*, från webbplatsen `http://w3.msi.vxu.se/users/rics/treebank/document/JSjobergh.pdf`, den 20 april 2004 (Gå in via `http://www.masda.vxu.se/~rics/treebank`, under länken Program, under rubriken session 4, under länken pdf 1)

[4] Carlberger/Domeij/Kann/Knutsson, *A Swedish Grammar Checker* (2002), från webbplatsen `http://www.nada.kth.se/theory/projects/granska`, den 22 april 2004

[5] Carlberger/Kann, *Implementing an efficient part-of-speech tagger* (1999), från webbplatsen `http://www.nada.kth.se/~viggo/papers.html`, den 16 augusti 2004

[6] de la Clergerie, Éric, *Appel à contribution sur la morpho-syntaxe* (2003), från webbplatsen `http://atoll.inria.fr/RNIL/TC37SC4-docs/slides070403.pdf`, den 17 juni 2004

[7] Clément, Lionel, *Language Resource Management Morpho – Syntactic Annotation Framework Draft French version* (2003), från webbplatsen `http://atoll.inria.fr/RNIL/TC37SC4-docs/N06.pdf`, den 25 juni 2004

[8] *CrossCheck – svensk grammatikkontroll för andraspråksskribenter*, projektbeskrivning på webbplatsen `http://www.nada.kth.se/theory/projects/xcheck`, den 14 augusti 2004

[9] *FreeText*, projektbeskrivning på webbplatsen `http://www.latl.unige.ch/freetext/en/description.html`, den 20 april 2004

[10] Granfeldt, Jonas, *Direkt Profil: Ett program för utvecklingsgångar och utvecklingsstadier i skriven inlärarfranska*, VR-ansökan, manuskript, Romanska institutionen, Lund

[11] Granfeldt/Schlyter, *Forskningsplan – Utvecklingsgångar i skriven inlärarfranska - från tal till skrift till datorprogram (USIF)*, forskningsansöknan, manuskript, Romanska institutionen, Lund.

[12] Heidorn, George, *Intelligent Writing Assistance* (2000), i "Handbook of Natural Language Processing", ed. Dale/Moisl/Somers, , kap 8, s 181-207

[13] Hobbs/Appelt/Bear/Israel/Kameyama/Stickle/Tyson, *FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text*, från webbplatsen `http://www.ai.sri.com/~israel/fastus-schabes-Discern.pdf`, den 12 augusti 2004

[14] Jensen/Heidorn/Miller/Ravin, *Parse Fitting and Prose Fixing* (1993), i "Natural Language Processing: The PLNLP Approach", ed. Jensen/Heidorn/Richardson, kap 5, s 53-64

[15] Kann/Domeij/Hollman/Tillenius, *Implementation aspects and application of a spelling correction algorithm* (1998), från webbplatsen `http://www.nada.kth.se/~viggo/stava/manual.html`, den 14 augusti 2004

[16] Knutsson/Pargman/Eklundh, *Transforming Grammar Checking Technology into a Learning Environment for Second Language Writing* (2003) från webbplatsen `http://www.nada.kth.se/~knutsson/knutsson_pargman_eklundh03.pdf`, den 22 april 2004

[17] Kostadinov/Thulin, *A text critiquing system for Swedish-speaking students of French* (2003), från webbplatsen `http://www.cs.lth.se/Education/Courses/EDA171/Reports/2003/jonas_fabian.pdf`, den 14 februari 2004

[18] Nivre, Joakim, *What kind of trees grow in Swedish soil? – A Comparision of Four Annotation Schemes for Swedish* (2002) från webbplatsen `http://www.msi.vxu.se/~nivre/papers/swedish_trees.ps`, den 20 april 2004

[19] Nugues, Pierre, *An Overview of Language Processing*, textkompendium, manuskript, Institutionen för datavetenskap, LTH, s 18-21

[20] Schlyter, Suzanne *Stades de développement en français L2* (1993), från webbplatsen `http://www.rom.lu.se/durs/STADES_DE_DEVELOPPEMENT_EN_FRANCAIS_L2.PDF`, den 15 augusti 2004

[21] Sjöbergh, Jonas, *Stomp, a PoS-tagger with a different view* (2003), från webbplatsen `http://www.nada.kth.se/~jsh/publications`, den 22 april 2004

[22] Vandeventer-Faltin, Anne, *Natural language processing tools for computer assisted language learning* (2003), från webbplatsen `http://www.linguistik-online.de/17_03/vandeventer.html`, den 16 augusti 2004

A Ordlista

ambiguitet Att ett ord kan ha flera olika betydelser. Ett svenskt exempel är *skär* som kan vara både adjektiv, substantiv och verb.

annotera Att märka upp en grammatisk struktur. Ett vanligt sätt att göra detta är med *taggar*.

böjd form Som uttrycket antyder en böjd form av ett grundord, *lemma*. Exempel: *är* är en böjd form av *vara*.

chunk Formeluttryck som visserligen går att böja, men som av inläraren lärts utantill och som därför inte tyder på regelapplikation. Icke att sammanblanda med *idiom*. Exempel: *Je m'appelle*.

CL/LT Akronym för Computational Linguistics/Language Technology.

finithet Personböjdhet. Huruvida ett verb är böjt efter person eller ej. *Parler* och *parlé* är ickefinita former medan *parle* och *parles* är finita former, böjda i första respektive andra person.

fullständig parsning Parsning där hela satser analyseras genom en flerstegsprocess, exempelvis innehållande tokenisering, PoS-taggning och så vidare.

förenat pronomen Pronomen som inte kan stå självständigt, utan ett verb. Exempel: *Je prends un café. Moi aussi.* Vi gör inte riktigt denna skillnad på svenska, både *je* och *moi* skulle översättas med *jag*. De två meningarna skulle alltså översättas med: *Jag tar en kaffe. Jag också.* Pronomenet *je* är förenat och kan inte stå utan ett verb.

genus Ett av de tre *särdrag* som definierar subjektspronomen i franskan. (De andra är *numerus* och *person*.) Det finns två genus; femininum och maskulinum. *Elle* (hon) är femininum och *il* (han) är maskulinum.

idiom En så kallad mängdfras (*set phrase*), det vill säga ett fast uttryck vars innebörd inte framgår av de ingående ordens betydelse. Icke att sammanblanda med *chunk*. Exempel: *Att kila vidare*

imparfait En böjningsform för förfluten tid i franskan som kommer ganska sent i utvecklingsgången för verbböjningar. Kan inte direktöversättas med svenskans *imperfekt*.

infinitiv Grundform av verb. *Vara* är grundformen till *är* och *gå* är grundformen till *gick*.

klitiskt pronomen Se förenat pronomen.

kongruens Överensstämmelse mellan två ord, exempelvis mellan verbet och dess subjektspronomen. Exempel: *Parles* (pratar) är böjt efter andra person singularis, varför strukturen *je parles* inte kongruerar eftersom *je* är första person singularis.

kopulaverb Verbet *är* i satser där verbet fungerar för att definiera eller beskriva något, snarare att det fungerar som hjälpverb eller rumsangivelse. Exempel: I satserna *Kalle är trevlig* och *detta är en katt* fungerar *är* som kopulaverb.

korpus Stor mängd autentiska texter som används som empiriskt material vid lingvistisk forskning.

lemma Ett ords grundform. Kan i vissa fall se helt annorlunda ut än vissa böjningar av det. Exempel: Böjningen *sämre*s lemma är *dålig*.

morfologi Formlära, läran om betydelsebärande orddelar. Exempel: *Talade* har två morfem, där *tala* är den lexikala betydelsen och *-de* har en grammatisk betydelse av förfluten tid. Båda är alltså betydelsebärande orddelar.

nominalfras En språklig enhet bestående av ett eller flera ord som tillsammans fungerar som subjekt i en sats.

numerus Ett av de tre *särdrag* som definierar subjektspronomen. (De andra är *genus* och *person*.) Det finns två numerus; singularis och pluralis. *Je* (jag) är singularis och *nous* (vi) är pluralis.

oförenat pronomen Se *självständigt pronomen*.

particip Icke-personböjd form av ett verb som i franskan typiskt används för att bilda *passé composé*-former. På svenska är particip de former som kan sättas efter *har*. Exempel: I satsen *jag har sprungit* är *sprungit* en participform.

partiell parsning En parsning som antingen endast gäller en del av en sats (så kallad lokal parsning) eller en parsning som endast innefattar vissa av de steg som utgör en *fullständig parsning*.

parsa Att analysera en sats grammatiskt och exempelvis ta ut satsdelar.

passé composé Den konstruktion för att uttrycka förfluten tid som franskinlärare lär sig först. Bildas genom presens av *être/avoir* och ett verb i particip. Exempel: *J'ai vu*.

person Ett av de tre *särdrag* som definierar subjektspronomen. (De andra är *genus* och *numerus*.) Det finns tre möjligheter för person; första, andra och tredje. *Je* (jag) är första person, *tu* (du) är andra person och *elle* (hon) är tredje person.

plus-que-parfait En böjningsform för förfluten tid i franskan som kommer ganska sent i utvecklingsgången för verbböjningar. Konstrueras genom *imparfait*-form av *être/avoir* + particip.

PoS Akronym för Part-of-Speech. Närmaste svenska översättning är ordklass, även om begreppen PoS och ordklass inte helt och hållet motsvarar varandra. PoS innehåller även information om *särdrag*.

PoS-tagga Att förse token i en text med en morfosyntaktisk tagg.

precision Procentmått som används för att ge en uppfattning om hur väl ett datalingvistiskt system klarar sina mål. Definieras som kvoten mellan antalet korrekt detekterade strukturer och det totala antal detekterade strukturer.

presens Verbform för nutid, pågående form.

pronomen Ordklass som i sin mest typiska användning står istället för substantiv. Pronomen kan också ersätta infinitivuttryck eller hela satser. Exempel: *jag, du, min, deras, henne, de, dem* och så vidare.

recall Procentmått som används för att ge en uppfattning om hur väl ett datalingvistiskt system klarar sina mål. Definieras som kvoten mellan antalet detekterade strukturer och det totala antalet strukturer.

självständigt pronomen Pronomen som står självständigt, utan verb. Exempel: *Je prends un café. Moi aussi.* Vi gör inte riktigt denna skillnad på svenska, både *je* och *moi* skulle översättas med *jag*. De två meningarna skulle alltså översättas med: *Jag tar en kaffe. Jag också.* Pronomenet *moi* är oförenat och kan stå utan ett verb.

starkt pronomen Se *självständigt pronomen*.

substantiv Ordklass innehållande ord som man kan placera artikel (sv. *en, ett*, fr. *le, la, les, des*) framför. Exempel: *båt, bil, kärlek, avstånd, trana, ved, tid* och så vidare.

syntaktisk Som har med syntax att göra, det vill säga i första hand ordföljd.

syntax I lingvistiska sammanhang innebär syntax grammatiska strukturer, deras konstruktion och deras respektive ordning. (Den datalogiskt orienterade läsaren är kanske mer van att ordet syntax används om programspråks uppbyggnad och de regler som bestämmer hur olika kommandon ska se ut och hur de kan kombineras.)

särdrag Den information som, förutom ordklassen, beskriver ett ords grammatiska egenskaper. Exempel: Ordet *parles* kan beskrivas genom att det är ett verb, böjt i andra person singularis. Informationen om böjningen i det här fallet är särdragen.

tagg En tagg är ett eller flera ord inneslutna i vinkelparenteser. Taggar används ofta i par för att innesluta den struktur man vill annotera. Exempel:

```
<span title="Pro.SG.P1">je</span>
```

token En enskild entitet i en ström av tecken, dock utan kommateringstecken. Dessa utgör nämligen egna tokens i vår bemärkelse av begreppet, se *tokenisering*.

tokenisering Att utifrån en ström av tecken skapa en ström av tokens.
Exempel: Meningen *Je m'appelle.* tokeniseras så:
token 1: *Je*
token 2: *m*
token 3: *'*
token 4: *appelle*
token 5: .

utvecklingsgångar Fenomenet att vissa grammatiska strukturer lärs in (och kan produceras) i en bestämd ordning. Man lär sig ofta behärska exempelvis presens innan man behärskar verbformer för förfluten tid i målspråket.

verb Ordklass innehållande ord som man på svenska kan placera *att* framför. Exempel: *avundas, blåna, springa, gå, hoppas, gilla, träna, blåneka, skala, grilla, hata* och så vidare.

XML Akronym för eXtended Markup Languge. XML är en standard för att annotera olika typer av information. Vilka taggar som tillåts och deras uppbyggnad definieras av användaren själv i en DTD-fil.

B Ur text.xml för Daniels exempeltext

```
<token id="token.0">je</token>
<token id="token.1">m</token>
<token id="token.2">’</token>
<token id="token.3">appelle</token>
<token id="token.4">adam</token>
<token id="token.5">et</token>
<token id="token.6">j</token>
<token id="token.7">’</token>
<token id="token.8">ai</token>
<token id="token.9">18</token>
<token id="token.10">ans</token>
<token id="token.11">. </token>
```

C Ur pos.xml för Daniels exempeltext

```
<tag id="tag.0" tok_id="token.0" lemma="je" pos_name="Pro.SG.P1"/>
<tag id="tag.1" tok_id="token.1" lemma="" pos_name=""/>
<tag id="tag.2" tok_id="token.2" lemma="" pos_name=""/>
<tag id="tag.3" tok_id="token.3" lemma="appeler"
                                  pos_name="Ver.IPre.SG.P1"/>
<tag id="tag.4" tok_id="token.4" lemma="" pos_name=""/>
<tag id="tag.5" tok_id="token.5" lemma="" pos_name=""/>
<tag id="tag.6" tok_id="token.6" lemma="je" pos_name="Pro.SG.P1"/>
<tag id="tag.7" tok_id="token.7" lemma="" pos_name=""/>
<tag id="tag.8" tok_id="token.8" lemma="avoir"
                                  pos_name="Ver.IPre.SG.P1"/>
<tag id="tag.9" tok_id="token.9" lemma="" pos_name=""/>
<tag id="tag.10" tok_id="token.10" lemma="" pos_name=""/>
<tag id="tag.11" tok_id="token.11" lemma="" pos_name=""/>
```

D Ur span.xml för Daniels exempeltext

```
<span id="span.0" from="token.0" to="token.3" rule_node="accord_lex"
                                  tag_name="verb_lex_with_accord"/>
<span id="span.1" from="token.4" to="token.5" rule_node="unknown"/>
<span id="span.2" from="token.6" to="token.11"
                                  rule_node="pres_accord_participe"
                                  tag_name="pres_accord_not_participe"/>
```

E Ur res.xml för Daniels exempeltext

```
<counter id="counter.0" counter_id="counter4h" counter_name="" value="4"/>
<counter id="counter.1" counter_id="counter4f" counter_name="" value="0"/>
<counter id="counter.2" counter_id="counter4e" counter_name="" value="0"/>
<counter id="counter.3" counter_id="counter4c" counter_name="" value="0"/>
```

```
<counter id="counter.4" counter_id="counter5h" counter_name="" value="0"/>
<counter id="counter.5" counter_id="counter5g" counter_name="" value="0"/>
<counter id="counter.6" counter_id="counter5e" counter_name="" value="0"/>
<counter id="counter.7" counter_id="counter5d" counter_name="" value="0"/>
<counter id="counter.8" counter_id="counter6e" counter_name="" value="0"/>
<counter id="counter.9" counter_id="counter6d" counter_name="" value="6"/>
<counter id="counter.10" counter_id="counter6b" counter_name="" value="4"/>
```

F Ur text.xml för Ritas exempeltext

```
<token id="token.0">j</token>
<token id="token.1">’</token>
<token id="token.2">ai</token>
<token id="token.3">commencé</token>
<token id="token.4">à</token>
<token id="token.5">étudier</token>
<token id="token.6">le</token>
<token id="token.7">francais</token>
<token id="token.8">à</token>
<token id="token.9">l</token>
<token id="token.10">’</token>
<token id="token.11">âge</token>
<token id="token.12">de</token>
<token id="token.13">treize</token>
<token id="token.14">ans</token>
<token id="token.15">.
</token>
```

G Ur pos.xml för Ritas exempeltext

```
<tag id="tag.0" tok_id="token.0" lemma="je" Ppos_name="Pro.SG.P1"/>
<tag id="tag.1" tok_id="token.1" lemma="" pos_name=""/>
<tag id="tag.2" tok_id="token.2" lemma="avoir" pos_name="Ver.IPre.SG.P1"/>
<tag id="tag.3" tok_id="token.3" lemma="commencer" pos_name="Ver.PPas.SG"/>
<tag id="tag.4" tok_id="token.4" lemma="" pos_name=""/>
<tag id="tag.5" tok_id="token.5" lemma="" pos_name=""/>
<tag id="tag.6" tok_id="token.6" lemma="" pos_name=""/>
<tag id="tag.7" tok_id="token.7" lemma="" pos_name=""/>
<tag id="tag.8" tok_id="token.8" lemma="" pos_name=""/>
<tag id="tag.9" tok_id="token.9" lemma="" pos_name=""/>
<tag id="tag.10" tok_id="token.10" lemma="" pos_name=""/>
<tag id="tag.11" tok_id="token.11" lemma="" pos_name=""/>
<tag id="tag.12" tok_id="token.12" lemma="" pos_name=""/>
<tag id="tag.13" tok_id="token.13" lemma="" pos_name=""/>
<tag id="tag.14" tok_id="token.14" lemma="" pos_name=""/>
<tag id="tag.15" tok_id="token.15" lemma="" pos_name=""/>
```

H Ur span.xml för Ritals exempeltext

```
<span id="span.0" from="token.0" to="token.3"
                    rule_node="pres_accord_participe"
                    tag_name="pres_accord_participe"/>
<span id="span.1" from="token.4" to="token.15" rule_node="unknown"/>
```

I Ur res.xml för Ritas exempeltext

```
<counter id="counter.2" counter_id="counter4e" counter_name="" value="7"/>
<counter id="counter.3" counter_id="counter4c" counter_name="" value="0"/>
<counter id="counter.4" counter_id="counter5h" counter_name="" value="0"/>
<counter id="counter.5" counter_id="counter5g" counter_name="" value="1"/>
<counter id="counter.6" counter_id="counter5e" counter_name="" value="0"/>
<counter id="counter.7" counter_id="counter5d" counter_name="" value="0"/>
<counter id="counter.8" counter_id="counter6e" counter_name="" value="0"/>
<counter id="counter.9" counter_id="counter6d" counter_name="" value="0"/>
<counter id="counter.10" counter_id="counter6b" counter_name="" value="0"/>
<counter id="counter.11" counter_id="counter4s" counter_name="" value="0"/>
<counter id="counter.12" counter_id="counter4r" counter_name="" value="0"/>
<counter id="counter.13" counter_id="counter4q" counter_name="" value="1"/>
<counter id="counter.14" counter_id="counter4p" counter_name="" value="0"/>
<counter id="counter.15" counter_id="counter4o" counter_name="" value="0"/>
<counter id="counter.16" counter_id="counter4k" counter_name="" value="1"/>
```

J Möjliga värden som barntaggarna till inflection kan anta

Grammatisk parameter	*Tagg*	*Attributvärde*
Genus	gender	feminine
		masculine
Numerus	number	sg
		pl
Person	person	1
		2
		3
Tempus	tense	present
		imperfect
		past
		future
Modus	mode	infinitive
		indicative
		participle
		conditional
		subjunctive

K Taggar som utökat regelformalismen

Tagg	*Attribut*	*Attributvärden*	*(Direkta) barntaggar*
description			
example			ex_found ex_notfound
ex_found			
ex_notfound			
recursive			
accord			criteriums category category
criteriums			criterium
criterium	value	gender number person	
category	value	noun verb adjective pronoun int_pronoun determiner adverb prePoSition conjunction numeral interjection abbreviation residual	

L Pronomen som är borttagna ur lexikonet

eux
la
laquelle
le
les
lui
leurs
leur
m
moi
personne
personnes
plusieurs
s

se
soi
toi
tous
tout
toutes

M Lemman vars uppslagsord tagits bort ur lexikonet

Orsak	*Lemma*
Substantiv som tolkas som verb	voiture
	chambre
	articles
Ord som inte finns	défonc
	révo
Ålderdomligt ord	icelui

N Pronomina utan information om person i lexikonet

auquel
aucun
ceci
cela
celui
certains
chacun
duquel
icelui
lequel
leur
mien
nôtre
nul
personne
plusieurs
quel
quiconque
sien
tel
tien
tous
vôtre

O Rader som saknar uppslagsord i lexikonet

Lemma	*PoS*
déséquilibrer	Ver:SPre+PL+P1
effleurir	Ver:SImp+PL+P1
effleurir	Ver:SPre+PL+P1
fritter	Ver:SPre+PL+P1
hennir	Ver:IPre+PL+P2
médicamenter	Ver:PPas
mezzo-soprano	Nom:Mas+PL
réapprendre	Ver:IPre+SG+P1
réapprendre	Ver:IPre+SG+P2
réentendre	Ver:SPre+PL+P1
réhabituer	Ver:SPre+PL+P2
teille	Nom:Fem+SG
teiller	Ver:Imp+SG+P2
teiller	Ver:IPre+SG+P1
teiller	Ver:IPre+SG+P3
teiller	Ver:SPre+SG+P3

P Översättningsnyckel för räknarnas olika id

Benämning i gränssnittet	*Id i regelfilen*	*Id i beslutsträdet*
Chunks	c01	—
Phrases sans verbe	c03	3a
Être/Avoir accord sujet/verbe	c11	4h
Être/Avoir pas accord sujet/verbe	c13	4i
Passé composé	c05	4c
Passé composé accord sujet/auxiliaire	c07	4e
Passé composé pas accord sujet/auxiliaire	c09	4f
Imparfait	c17	4k
Imparfait accord sujet/verbe	c21	4q
Imparfait pas accord sujet/verbe	c23	4r
Autre temps de Être/Avoir	c22	4s
VModal accord sujet/verb (présent)	c06	5d
VModal pas accord sujet/verb (présent)	c08	5e
AuxMod + Infinitif	c15	5b
AuxMod + Infinitif accord sujet/aux	c02	5g
AuxMod + Infinitif pas accord sujet/aux	c04	5h

Verbe lexical non conjugué	c10	6b
Verbe lexical conjugué et accord	c12	6d
Verbe lexical conjugué pas accord	c14	6e
Verbes lexicaux conjugués	c16	7
Plus-que-parfait	c19	4m
Plus-que-parfait accord sujet/auxiliaire	c18	4o
Plus-que-parfait pas accord sujet/auxiliaire	c20	4p

Q Chunks som preliminärt inkluderats i chunksfilen

je m'appelle
je voudrais
s'il vous plaît
je ne sais pas
c'est
qu'est-ce que c'est
il y a
il faut
il n'y a pas
j'ai mal

R Testtexter

R.1 Hans

C'est un trés chaud jour au été. Il y a un garçon sur une petite île. Il s'appeler Michel. Il y a une vert maison avec un bleu porte, le garçon habiter a la maison. Le garçon aimer le fleures avec la île. Il arrive á la île dans un petite bateau. Un jour quand Michel il vouloir builder un bridge au une different île, arrive deux garçons. Les garçons builder un ville dans la different île. Le ville est trés grosse. Michel est malheurese, parce que le voisin-île est trop de grosse.

Nivå: 1
Totalt antal strukturer: 6
Antal detekterade strukturer: 6
Antal korrekt detekterade strukturer: 5

Struktur	*Taggat som*	*Skulle taggats som*	*Trolig felkälla*
il arrive	Verbe lexical conjugué pas accord	Verbe lexical conjugué accord	Ordningen i lexikonet

R.2 Roland

1. C'est un faire de soleil, la maison vert. Les fleurs rouge, plage vert, ciel bleu.
2. Bonjour je m'appelle Roland et j'ai un probleme.
3. Je habite un a plage avec moi, je déteste ca!
4. Je travaille chaque jour
5. Un jour il va aller á la maison blanche
6. Bonjour, Lennart
- Bonjour!
7. un cafe?
8. OHH..lala!! un café, trés bien.
9. les mobylettes, ils vont du cafe?
10. 32 jours.....un cafe et un l'hopital
11. 62 jours....un cafe, un l'hopital, en bar et un hotel
12. OHH, un VILLAGE
13. J'ai petit maison
14. j'ai déteste ca!
15. Host Host
16. Lennart vais un "boat"
17. AHH. le gross soleil

Nivå: 1
Totalt antal strukturer: 10
Antal detekterade strukturer: 10
Antal korrekt detekterade strukturer: 10

Kommentar: Trots att detta är en text på språknivå 1, hittas inga satser utan verb. Det beror på att texten inte innehåller några satser med pronomen, men utan verb. Däremot finns ju satser med interjektioner och andra typer av subjekt än pronomen. Dessa är dock inte heller meningen att Direkt Profil skall detektera i nuläget.

R.3 Lillie

L'homme sur l'île. Il fait du soleil. Un ciel blue. Une maison vert. Une six fleur lila. Un plage vert. Un homme. Un pantalons orange, une pull noir. Un chair (stol) rouge et blanc. C'est un faire du soleil. Le homme a un mustache noir. Un petit maison. Le rouge. Un trois fleur lila. Un plage vert et petit (gult) Un chien deux. Un blanc et noir. Les trois hommes. Les trois hommes a parle le francs et une usines. Le homme regader la usine. Les six hommes. Une cinq mobylette la vert, rouge, orange, blanc et noir. Une petite maison, une usine et une grosse maison. Les maisons (blir) grosse et grosse. Les maisons (blir) grosse et GROSSE. Un petit village. Le hôtel et un bar. Les cinq hommes. Une deux voitures blanc et vert. Le homme (blir galen). Le homme aime ne voitures pas. Un homme. Une grosse soleil (gul). Un homme. Une petite maison blanc.

Nivå: 1
Totalt antal strukturer: 2
Antal detekterade strukturer: 2

Antal korrekt detekterade strukturer: 1

Struktur	*Taggat som*	*Skulle taggats som*	*Trolig felkälla*
il fait	VModalpas accord	VModal accord	Ordningen i lexikonet

R.4 Lena

Il y a un homme vieux avec un moustache marron. Il s'appelle Paul. Il voit la mer qui il fait chaque matin. Il a une maison petite. La maison a un porte bleu. Sa jardin a des fleurs rouges. C'est un beau jour, le soleil brille, il fait chaud. Le ciel est clair et bleu. Il est seul sur l'île. Il est triste parce qu'il a ne personne que parler avec. Les jour passer et il a ne fait rien. Des oiseux n'aime pas lui. Il est seul. Il veut construire un doc à son île. Des messieurs et de leurs chiens arrivont à son île. Il est content. Ils devenir copains. Les hommes proPoSer une proPoSition. Paul accepté la proPoSition si construire une usine sur l'île. Finalement Paul devenu trés content avec sa vie. Plus gens venir à son île. Paul devenu plus trés content. Des gens veux construire plus maisons et fabriques. Finalement devenue l'île une ville! Pollutions et plus des gens. Paul devenu triste. Il n'aime pas la ville. Il prend son bateau et va à un île noveau. C'est calme et maintenant est il vraiment content avec son vie!

Nivå: 2
Totalt antal strukturer: 20
Antal detekterade strukturer: 20
Antal korrekt detekterade strukturer: 16

Struktur	*Taggat som*	*Skulle taggats som*	*Trolig felkälla*
il s'appelle	Verbe lexical pas accord	Verbe lexical accord	Ordningen i lexikonet
il fait	V Modal pas accord	V Modal accord	Ordningen i lexikonet
il fait	V Modal pas accord	V Modal accord	Ordningen i lexikonet
il n'aime	Verbe lexical pas accord	Verbe lexical accord	Ordningen i lexikonet

Kommentar: Strukturen *il vraiment content* markeras, vilket är korrekt eftersom *content* står upptaget i lexikonet som en böjning av lemmat *conter*. Detta är dock inte det önskvärda på sikt i Direkt Profil.

R.5 Thea

C'est une petite île verte. Dans cette île il y a une maison verte seulement avec une fenêtre. Des fleures dans l'île et très belles et rouge. Le ciel est bleu et la mere est aussi bleue. Le soleil est jeune et bruille. C'est une homme avec un chapeau que met à STOL. C'est une bateau blanche près l'île. Deux oiseaus sont OVANFÖR l'île. Les oiseaus ont arrivé à l'île et l'homme,

qui s'appelle Pierre, dors. Les oiseaus disent des mot d'oiseau et Pierre se reveille. Pierre va faire un point de sa île et une autre île. La point est prête et il va à la point. Deux amis de pierre arrivent avec leures deux chiens pour rencontre pierre. Un chien est noir et un est blanc. Les hommes ont des chapeaus noirs et des lunettes. Ils disent que ils veulent faire un usine à l'île de Pierre et ils le vont donner beaucup des argent. Pierre dit que c'est OK et maintenant il a un grand usine à son île. Des hommes de travailles arrivent à leures velos. Maintenant c'est une grande maison au bord la maison de pierre. Mais regarde! C'est une petite belle ville maintenant avec un cathedral dans le centre de l'île. C'est une grande île maintenant et ne plus une ville, c'est un village avec des maison très hautes! C'est Pierre avec sa petite maison et les deux oiseaus au centre dans le village. Mais Pierre n'aime pas le circulacion. Et il et les oiseaus n'aiment pas le fume. La soir, Pierre et les oiseaus prendent le bateau et vont à une nouvelle île. La lune est très grande et bruille à Pierre que fait la peche et amie la vie.

Nivå: 3
Totalt antal strukturer: 16
Antal detekterade strukturer: 16
Antal korrekt detekterade strukturer: 16

Kommentar: Strukutren *il et les oiseaus n'aiment pas* taggas som mening utan verb på grund av att fönsterstorleken är satt till fem. Detta är alltså korrekt parsat enligt de regler som Direkt Profil har att arbeta med. Även om fönsterstorleken ökas skulle detta taggas som icke-kongruens, eftersom kongruens kontrolleras mellan *il* och *aiment*.

R.6 Amy

Quand je suis été 12 ans j'ai choisi l'langue de France pour étudier dans l'école. J'ai étudié dans Kulladalsskolan, une trés bonne l'ecole, mon professeur a été trés bien et j'ai aimé l'langue trés vite. J'ai étudié francais plusieurs anneés dans l'école mais j'ai eu aussi étudié francaise dans la France. Par example, quand je suis été 13 ans j'ai fait une voyage de une semaine avec ma mére, nous sommes allé a Paris. Nous avons vu l'tour Eiffel, Le Louvre, Triumf bågen et beacoup de autre choses. J'ai eu un problem quand je suis été a Paris, le francoises elles a parlé trés vite et je ne ai rien compris, mias j'ai appris parle francaise une petite peu. C'est été une trés bonne semaine et je ne le oubliais jamais. Quand je suis été 15 ans, je suis allé à la France avec STS language school je suis été dans Canne et aussi dans Juan les pins pour trois semaines. Nous avons eu lecons toutes les jours (pas samedi et dimanche) et nous avons étudié seulement francaise (naturellement). Nous avons eu un professeur qui ne a pas parlé suedoise et une professeur qui a parlé suedoise et francaise. Quand nous avons eu une lecon avec le professeur qui ne a pas parlé suedoise nous avons du parler francaise, il n'a pas parlé englaise. C'est été trés difficile mais cette fois j'ai appris parlé francaise bien.Maintenat j'étudier francaise dans Öresundsgymnasiet. Nous avons trois lecons par semaine,ce n'est pas souvent et je voudrais étudier toutes les jours comme j'ai fais à la France (mias pas samedi et dimanche). Dans toutes mes classes à francaise vieilles et novelles, c'est été classes avec peu de eleves. J'pense c'est trés bien.J'aime la France et je voudrais parler francais coulant aussi écris coulant. Mais francaise est une trés difficile langue.

Nivå: 3
Totalt antal strukturer: 37

Antal detekterade strukturer: 37
Antal korrekt detekterade strukturer: 37

Kommentar: Participet *étudié* står inte med i lexikonet som verb. Det står däremot med som adjektiv. Det förekommer på tre ställen i texten; två gånger i stuktueren *j'ai étudié* och en gång i *nous avons étudié*. Dessa taggas endast som *j'ai* respektive *nous avons*, vilket ju är rätt givet lexikonets information. På sikt är vore det naturligtvis önskvärt att utöka lexikonet för att undvika detta problem.

R.7 Nellie

J'ai commencé apprendre le français quand j'allais en le 6e class. J'ai parlé le français en 4 ans et j'ai le parle / je le parle / assez bien. J'ai facile pour parler le français et je pense c'est très gaie. J'ai été en France pluesieurs de fois. Cet été j'ai été à Juan-les-Pins en France pour 3 semaines avec mes amies. C'était très dur parce qu'avant l'été je n'étais pas assez bien de parler le français. Tous ils ont me dit je comprenais seulement 30-40 pout cent, mais maintenant je comprends presque de tout. Il est parce que j'ai une proffeseuse que parle le français très vite. C'est très dur! Je parle l'espagnol aussi et parce que c'est assez dur avec foccuser de français mais il sort. Je parle aussi l'anglais et j'ai parlé l'anglais à 6 ¡ ans maintenant mais je le pense est très facile. Bueacuop de facile que le français ou l'espagnol.Quand j'étais une enfant ma famille et moi, nous avons eu sur une voyage en France presque chaque année. J'adore La France et j'aime tout de France, alors, presque de tout. je n'aime pas les hommes de français. Ils sont (snuskiga och äckliga). Ils pensent nous, des filles de Suede, voulons coucher avec eux. Mais il est ne correct pas.

Nivå: 3
Totalt antal strukturer: 33
Antal detekterade strukturer: 33
Antal korrekt detekterade strukturer: 33

Kommentar: Strukturen *nous , des filles de suede* taggas som mening utan verb på grund av att fönsterstorleken är satt till fem. Detta är alltså korrekt parsat enligt de regler som Direkt Profil har att arbeta med.

R.8 Ella

Dans la première image, il y a un homme qui habite dans une maison verte. Sa maison est très petit et la porte est bleue. L'homme porte un chapeau blanc et il est assis dans une chaise. Dans l'île, il y a des fleurs. Elles sont rouges et très belles. L'homme a un petit bateau aussi. La mer est très calme et la soleil brille. Il y a deux oiseaux dans la ciel.Cet histoire commence un beaux jour: la soleil brille, et il y a deux oiseaux dans la ciel bleue. L'homme dort dans sa chaise, mais soudain, les deux oiseaux lui reveille. /Ils le réveillent. / Les oiseaux commence a parler avec l'homme, et ils lui demande s'il n'est pas seul dans ce petit île. L'homme il réponds: Oui, mais j'ai mon bateau et je peux visiter mes amis qui habitent dans la ville si je veux.Les oiseaux, qui s'appellent Jean et Pierre, pensent que l'homme doit construire un petit pont. L'homme pense que c'est un bon idée, et il commence tout de suite. Le pont est très beaux et après seulement

un jour, deux messieurs sont allés pour visiter l'homme. Ils disent que l'homme doit construire une usine, et gagner beaucoup d'argent.C'est un bon idée, l'homme pense. Et trois semaines plus tard, l'usine est terminé. Dans l'usine, cinq hommes travaillent. Mais ils ne pensent pas que c'est drôle qu'ils doivent faire la velo pour aller au travail. L'homme le comprend et avec l'aide des hommes, il construit un grand maison où les hommes peuvent habiter. Mais l'histoire n'est pas terminé ici. Les hommes ont le besoin d'acheter la nourriture etc. Un homme est très religeuse et il voudrait une église. L'autre homme voudrait un cinema et le troisième voudrait une coiffeuse. L'île est plus et plus grand. Il y a un banque, un hotel etc. L'homme est assis comme d'habitude avec Jean et Pierre près de sa maison, mais il ne peut pas entendre qu'ils disent à lui. Il y a trop beaucoup de bruîle! L'homme est devenu complètement fou sur l'île. Il n'aime pas les voiture et tous les gents qui habite dans son île. Il prend son bateau et il est allé avec Jean et Pierre. Ils cherchent un nouveau île où l'homme construe une nouvelle maison. La bas, il peut faire la pèche, et seulement être assis pour voir la belle mer. L'homme il a appris un devoir très important: On ne doit pas faire des choses stupides seulement pour l'argent!

Nivå: 3
Totalt antal strukturer: 33
Antal detekterade strukturer: 33
Antal korrekt detekterade strukturer: 30

Struktur	*Taggat som*	*Skulle taggats som*	*Trolig felkälla*
il commence	Verbe lexical pas accord	Verbe lexical accord	Ordningen i lexikonet
il n'aime	Verbe lexical pas accord	Verbe lexical accord	Ordningen i lexikonet
il construit	Verbe lexical non conjugué	Verbe lexical conjugué et accord	Ordningen i lexikonet

Kommentar: Strukturen *ils lui* parsas som icke-personböjt lexikalt verb. Det beror på att *lui* tolkas som particip av lemmat *luire*. Det "riktiga" verbet *demande* (icke-kongruens) missas därigenom. Det enklaste sättet att komma till rätta med ett sådant här problem är att ta bort *luire* ur lexikonet. Fördelarna väger klart tyngre än nackdelarna med att göra det, eftersom inlärarna sällan eller aldrig använder sig av det verbet.

R.9 Christine

1998, j'ai allé á Paris avec ma classe. Beaucoup de personnes, je pense 20-22 peut-être.C'était ma prof de francais, mes amies dans ma class francais, ma mère, et des êleves dans une autre école.Pour faire cette visite, il faut beaucuop d'argent. Mais, en Mai, 1998, nous allons là, á Paris. Une ville trés magnifique.À Paris, nous avons habité á beucoup de familles. Ma famille, avec ma fille"était trés gentile.Ils me donnerais de chooses á manger tout le temps.Je pense, cette famille était un peu riche, parce-que le père était la principal dans l'école. À Paris, nous avons visité des places trés connu. Naturelment, l'arc de triouphe, la tour Eiffel, le Notre Dame et aussi beaucoup des autre chooses connu á Paris. J'aimerais bien la Sacre-Couer. C'est un église magnifique, je pense. Dans la nuit á Paris, il y beaucoup de personne qui est dans l'éscalier devant la Sacre-Couer.Ils chanterent et boivent beaucoup de bière, tout le temps. Dans la nuit,

Paris était une ville magnifique, avec des personne et le musique.Nous resterons á Paris pendant une semaine. Je pense ma fille, je ne connais le nom maintenant, était une fille trés gentille.Elle avais une souer, une mère et un pére, aussi un chien, très gentil. J'ai beaucoup de fotos de ma visite.

Nivå: 3
Totalt antal strukturer: 19
Antal detekterade strukturer: 19
Antal korrekt detekterade strukturer: 19

Kommentar: Strukturen *nous avons habité* märks upp som *être/avoir* med accord eftersom *habité* endast står som adjektiv i lexikonet och inte som verb. Endast en mening utan verb hittas, nämligen *il y beaucoup de personne qui est dans*, vilket inte är vanligt på den här nivån. Den troliga färklaringen är att författaren har bara glömt *a*. Dock skall sägas att Direkt Profil parsat denna struktur korrekt givet dess nuvarande utseende.

R.10 Nicole

Moi, je me suis toujours intéressée aux langues. Même la grammaire m'interesse, ce qu'on trouve ennuyant normalement. ça a tout commencé il y a sept ans, au collège. J'ai décidé d'étudier le français plutôt que l'allemande parce que je trouvais l'allemande une langue trop moche. J'ai eu un prof qui a tout á fait accroché mon intêret pour le francais. Pendant les lecons on a chanté, on a regardé des films, on a invité des francais pour nous parler de la vie francaise. Bref, il nous a vraiment engagés. C'est ainsi alors que j'ai fait le decision de continuer mes études de francais au lycée. Je me suis inscrite dans une programme de francais dans laquelle on dédie tout son choix individuel pour le francais et j'ai appris vraiment beaucoup. Au seconde classe on a fait une échange scolaire avec une classe francaise à Nevers. J'ai habité chez une famille francaise et j'ai fréquenté l'école francaise. C'était une grande experience! La même année moi et mon amie Anna nous avons decidé de chercher de trouver un boulot en France pendant l'été. Alors on a contacté une couple suédoise d'un certain âge qui habite près de Poitiers. Cette couple elle nous a donné l'adresse à un château. On y a écrit et voilà on a eu la permission d'y aller pour habiter et pour donner un main à Madame Ogilvy. Ca a été une souvenire extraordinaire. Un grand château, un jardin ouvert pour les touristes, des personnes fameauses etc..etc. Rentrées en Suède, au terminal, j'ai écrit mon travail special en francais. Je l'ai intitulé. L'amour de la perspective de Françoise Sagan. C'était une composition qui voulait beaucoup de temps à écrire mais qui était aussi très intéressant d'écrire. J'ai lu Bonjour Tristesse, Les nuages merveilleux et des autres livres de l'auteur et j'ai réflêchi beaucoup sur les textes de Sagan. Après avoir terminé au lycée j'ai laissé tomber un peu le francais. Tout simplement j'ai fait des autres choses. J'ai fait des études a l'univeresitet et j'ai été en Italie. Mais le francais a toujours resté dans mon coeur. J'ai regardé des films francais, recamment Pola X et j'ai passé quelque semaine en France. Par quelques amis francais j'ai entretenu mes liaisons avec la langue sans l'oublier completement. Dès que j'ai réalisé que je devrais étudier le francais à l'université cet printemps je me suis mise à lire les belles lettres en francais de nouveau. J'ai commencé avec une livre plus facile a lire, La Plage pour m'habituer à la langue. A ce moment j'ai presque atteint a la fin du livre et je me sens prêt à commencer mes études en réal. ça sera vraiment interessant!

Nivå: 4
Totalt antal strukturer: 43
Antal detekterade strukturer: 43
Antal korrekt detekterade strukturer: 41

Struktur	*Taggat som*	*Skulle taggats som*	*Trolig felkälla*
on trouve	Verbe lexical conjugué pas accord	Verbe lexical conjugué et accord	Ordningen i lexikonet
on dédie	Verbe lexical conjugué pas accord	Verbe lexical conjugué et accord	Ordningen i lexikonet

Kommentar: Participet *réflêchi* är felstavat[50], och hittas därför inte i lexikonet. Resultatet blir att endast *j'ai* taggas i lexikonet. Strukturen *nous parler* taggas som icke-personböjt lexikalt verb vilket är korrekt eftersom Direkt Profil inte kan veta att *nous* i det här fallet är objektspronomen. Det tolkas därför som subjektspronomen och det efterföljande verbet taggas.

R.11 Ingmar

Il était une fois 2 femmes qui s'appelaient Sylvie et Simone. Ils avaient travaillé toutes ses vies pour pouvoir faire un voyage en Italie. Sylive, une dame assez grosse avec les cheveux blondes qui aimait se dresser en verte, voulait partir en Italie pour se réPoSer au soileil à la plage est devenir broncée. Simone, une fille mince avec des cheveux brunes et longues qui aimat le coleur bleu, voulait y aller pour faire la conaissance des beaux mecs italiens. Une belle journée, ils sont parties. Ils vont voyager par la voiture verte de Sylvie, (naturalement). Quand ils étaient en train de mettre ses pinales à la voiture, c'était très dûr parce que ils avaient beaucupr de valises. Mais une heure plus, ils ont laissé la Suède pour aller en Italie par l'autoroute. Après deux jours, ils sont arrivées en Italie. Ils veulent vivre à l'hôtel Magnifico et ils ont reçu la clé pour leur chambre du receptioniste, Carlo. La chambre était merveilleuse. Elle donnait sur la plage, et on pouvait voir l'eau bleu et le soleil qui brillait. Sylvie admirait la vue autant que Simone s'installait dans la chambre. Ils avaient seulement une chambre mais il y avait deux lits très comfortables. Sylvie et Simone était très jolies pour avaient trouvé leur paradis. Sylvie et Simone s'amusaient beaucoup. Elles passaient ses jours à la plage, au soleil. Ils ont vu des voiles sur l'Océan et des oiseaux au ciel. C'était fantastique, un vrai paradis! Elles faisaient la conaissance de la cuisine italienne aussi, à leur restaurant favorite, Taverna di Luigi. Le soir, elles aimaient être au bar, ou elles ont bu du vin et parlaient avec le bartender, Pablo. Un soir, elles ont vu deux hommes qui sont entrés. Les garcons s'appelaient Don et Juan et ils ont devenus très gaies d'y trouver deux jolies filles. Ils ont achetés l'alcohol et tous les quatres ont bus ensemble. Sylvie s'est amusée avec Don et Simone a fait la conaissance de Juan. Ils ont dancé toute la nuit!! Le lendemain, ils ont fait un tour de sightseeing en Barcelona. Un guide ont raconté des histoires très intèressantes de la vie et les gents de la ville. Le soir, Sylvie et Juan se sont décides de voir le soileil qui descendait à neuf heures. Ils le voyaient en silence, mais il ne fallait pas des mots parce que ils savaient que ils avaient trouvés le grand amour. Simone et Don sont allés au Ristorante Vita Bella. Un homme a joué le guitar et il a chanté très bon. Simone et Don voudraient célebrer qu'ils ont décidé de se marier! Ils sont tombés amoureux, tous les deux. La semaine était finie à toute vitesse. Sylvie et Simone ne veulaient

[50]Det ska stavas *réfléchi*.

pas rentrer en Suède, mais il devaient. Elles étaient très tristes de laisser leurs amours noveaux. Mais quand elles ont pris leurs valises pour partir, Don et Juan ont couru vers la voiture. - Attendez-nous! Nous voulons vous accompagner à la Suède! Don a diré: - Simone, je t'aime! Et Juan a chanté: - Sylivie, ma cherie, je t'adore! Ensuite, ils sont partis ensemble à la voiture verte pour commencer une nouvelle vie en Suède ensemble. Et ils étaient très heureux. Tout est bien qui finit bien!

Nivå: 4
Totalt antal strukturer: 31
Antal detekterade strukturer: 31
Antal korrekt detekterade strukturer: 31

Kommentar: Strukturen *ils sont parties* är inkorrekt eftersom *parties* är böjt efter *elles* och inte *ils*. Dock är det inte inbyggt i beslutsträdet att kongruenskontroll ska ske mellan particip och dess pronomen, som tidigare nämnts i avsnitt 4.1 på sidan 24. Strukturen *ils ont dancé* taggas inte eftersom *dancé* inte finns med i lexikonet; endast *ils ont* taggas därför. Vidare kan nämnas att *nous ! nous voulons vous accompagner* skulle ha markerats som *nous voulons vous accompagner* om satsavgränsningar varit implementerade. Nu behandlas utropstecknet som ett (okänt) ord. Slutligen ska nämnas att en felaktighet när det gäller hur *imparfait*-räknarna räknas upp kan ha upptäckts i och med denna testkörning; räknaren *imparfait* ska utgöra summan av de två andra *imparfait*-räknarna och har efter parsningen värdet 11. Den första av dessa har värdet 9 och den andra 0. Det visar sig att 9 och 0 stämmer; så många strukturer av respektive räknare finns taggade i texten.

R.12 Inga

J'étais en 7ème et j'avais 13 ans. J'aimais beaucoup la France, et voulais certainement apprendre le français. Le prof étais sympa et on passait des leçons très agréable. Il n'y avait pas beaucoup des élèves et on apprenait vite. Les premieres mots c'étaient le jour de la semaine, oui et non et des trucs assez simple. J'aimais bien aller en cours en collège. J'aimais bien le français, et j'étais interéssée, et les cours n'étaient pas seulement des leçons d'école. Quand j'avais 16ans j'ai commençais le lycée. / j'ai commencé le lycée ./ J'aimais toujours la France comme pays, mais je suis devenue assez lasse à l'école, en cours de français aussi. On faisait presque rien pendant les cours ou bien on faisait des choses très ennuyants et les élèves trouvaient ça nul.Mais comme je ne voulais plus travailler à la maison (il y avait trop des devoirs!!!) je décidais de passer un an d'ailleurs. Je voulais quitter la Suède qui m'ennuyait. Je me souvient excatement comment c'était pris, la décision. Pendant un repas avec la famille j'ai crié avec haute voix, en mangant que j'allais aller en France pour faire des études. Mes parents étaient contentent. J'ai pris l'avion le premier Septembre 2000. J'étais hyper nerveuse parce que j'allais habiter chez une famille française toute l'année! Mais, bon c'était très bien, ils étaient très gentilles avec moi. Je suis allée au lycée français et ils m'ont mis en premier L (littéraire) un. C'est à dire 6 heures de français par semaine!!! C'était très dur, mais j'ai beaucoup appris. Non seulement la lange mais plein des choses dans la littérature française. bon, on doit términer. J'espère que ça vous a aidé.

Nivå: 4
Totalt antal strukturer: 38

Antal detekterade strukturer: 38
Antal korrekt detekterade strukturer: 38

Kommentar: Strukturen *nul. Mais comme je ne* markeras som sats utan verb. Med Direkt Profils nuvarande implementering är detta korrekt; nul är markerat som pronomen i lexikonet och därför ska en strukturtaggning börja här. Satsavgränsare är ju ännu inte implementerat i Direkt Profil och därför behandlas punkten som vilket (okänt) ord som helst. Strukturen *je décidais de passer* parsas som icke personböjt verb. Det beror på att *décidais* inte är en korrekt form och inte hittas i lexikonet. Direkt Profil fortsätter därför inom fönstret, det vill säga de närmsta fem orden och letar efter ett verb. Detta hittas i *passer* som ju är icke-personböjt. Strukturen *on doit términer* är markerad som modalverb utan efterföljande infinitiv. Det beror på att *términer* är felstavat (skall stavas *terminer*). Strukturen *vous a aidé* är markerat som icke-kongruens, vilket stämmer eftersom det föregående *ça* är borttaget ur lexikonet. Här är alltså ett exempel på när ett sådant val är en nackdel i Direkt Profil. Hade *ça* funnits kvar hade denna struktur parsats som *passé composé* med kongruens, vilket gett en mer rättvisande bild åt denna texts språknivå. Valet att tillfälligt ta bort *ça* ur lexikonet grundar sig på att det verkar som om fler strukturer blir korrekt parsade utan *ça* i lexikonet. Även i denna text, liksom i Ingmars text, verkar *imparfait*-räknarna felaktiga; summan är 12, men de två delarna är 10 respektive 1. Det är dessa sistnämnda två värden som motsvarar antalen uppmarkerade *imparfait*-strukturer i texten.

S Räknarna i webbgränssnittet

De räknare som har TAGGAS INTE som *Taggningsexempel* färgmarkeras och räknas inte separat (med undantag för imparfait som räknas) eftersom de utgör summor av två andra, mer specifika, räknare.

Benämning i gränssnittet	*Förklaring*	*Taggningsexempel*
Chunks	Chunks. Ännu inte fullt ut implementerat.	Je voudrais un chien.
Phrases sans verbe	Satser utan verb	Je á Paris en train.
Être/Avoir accord sujet/verbe	Être/Avoir i presens med kongruens mellan subjekt och verb	J'ai 13 ans.
Être/Avoir pas accord sujet/verbe	Être/Avoir i presens utan kongruens mellan subjekt och verb	Tu est gentil.
Passé composé	Passé composé-konstruktion	*TAGGAS INTE*
Passé composé accord sujet/auxiliaire	Passé composé-konstruktion med kongruens mellan subjekt och hjälpverb	J'ai mangé bien.
Passé composé pas accord sujet/auxiliaire	Passé composé-konstruktion utan kongruens mellan subjekt och hjälpverb	Tu a mangé bien.
Imparfait	Être/Avoir i imperfekt	*TAGGAS INTE*

Imparfait accord sujet/verbe	Être/Avoir i imperfekt med kongruens mellan subjekt och verb	Tu étais lá?
Imparfait pas accord sujet/verbe	Être/Avoir i imperfekt utan kongruens mellan subjekt och verb	Nous avait une voiture.
Autre temps de Être/Avoir	Être/Avoir i andra tempusformer än presens och imperfekt	Je sera heureuse.
VModal accord sujet/verb (présent)	Modalt verb i presens med kongruens mellan subjekt och verb	Elles font ses devoirs.
VModal pas accord sujet/verb (présent)	Modalt verb i presens utan kongruens mellan subjekt och verb	Elles faut ses devoirs.
AuxMod + Infinitif	Modalt hjälpverb med efterföljande infinitiv	*TAGGAS INTE*
AuxMod + Infinitif accord sujet/aux	Modalt hjälpverb med efterföljande infinitiv med kongruens mellan subjekt och hjälpverb	Tu sais parler anglais?
AuxMod + Infinitif pas accord sujet/aux	Modalt hjälpverb med efterföljande infinitiv utan kongruens mellan subjekt och hjälpverb	Tu veut avoir un cadeau?
Verbe lexical non conjugué	Icke personböjt lexikalt verb	Le soir, je boire du lait.
Verbe lexical conjugué et accord	Personböjt lexikalt verb med kongruens mellan verb och subjekt	Je mange avec toi.
Verbe lexical conjugué pas accord	Personböjt lexikalt verb utan kongruens mellan verb och subjekt	Tu mange avec moi.
Verbes lexicaux conjugués	Personböjda lexikala verb	*TAGGAS INTE*
Plus-que-parfait	Pluskvamperfekt-konstruktion	*TAGGAS INTE*
Plus-que-parfait accord sujet/auxiliaire	Pluskvamperfekt-konstruktion med kongruens mellan subjekt och hjälpverb	Tu avais mangé bien?
Plus-que-parfait pas accord sujet/auxiliaire	Pluskvamperfekt-konstruktion utan kongruens mellan subjekt och hjälpverb	Nous avait vu le circus.

T Exempeltext Daniel

Je m'appelle Daniel et j'ai 18 ans. Je suis en terminal à la école. Je commencer lire francais dans six grade. Ma mère as habit en France. Nous voyager à Paris deux. J'aime parle francais. Dans l'école je ne apprendre rien francais. Sur quelque ans je voyager au fran-

cais Martinique et apprendre parle francais. À maison je parfois regarde un film francais. Je ne ai pas de francais copains. Pour a lire francais trois ans je ne pas beaucoup parle francais. J'aime parle francais, beaucoup. Je crois un voyager à Paris c'est tres bon pour étudie francais.

U Exempeltext Rita

J'ai commencé à étudier le francais à l'âge de treize ans. Je me suis toujours intéressée aux langues, et j'ai trouvé le francais une langue très belle. Au début la beauté de la langue était la raison de mon intérèt. Il n'a pas pris longtemps avant que j'ai trouvé que des études d'une langue étrange m'a donné beaucoup plus que beauté. En lisant des textes et livres en francais, je me suis sentie comme si j'étais dans un monde completement différent. En parlant j'ai découvert un sentiment de comment il doit être pour un enfant qui apprend sa langue maternelle.