

Experiment inom samsyftningsresolution

CECILIA ROES
och RICHARD HELSING



**KTH Datavetenskap
och kommunikation**

Experiment inom samsyftningsresolution

CECILIA ROES
och RICHARD HELSING

Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik
Kungliga Tekniska Högskolan år 2010
Handledare på CSC var Johan Boye
Examinator var Mads Dam

URL: [www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2010/
roes_cecilia_OCH_helsing_richard_K10091.pdf](http://www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2010/roes_cecilia_OCH_helsing_richard_K10091.pdf)

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Sammanfattning

Co-reference resolution is a subproblem within language technology which originated from when scientists first tried to translate texts with computers. There are some suggested algorithms and the purpose of this project is to see how well one of those algorithms can find anaphors in a given text compared to humans. To find an answer to this question a survey was made which includes several texts. Some of them were picked randomly among literary, academic and news article texts and others specifically created. The people participating in the survey were chosen randomly and have different knowledge about linguistics. The result show that co-reference resolution is a trivial problem for humans while it's a very hard one for computers. There is still a lot of work to be done in the field.

Innehåll

1	Inledning	1
2	Bakgrund	2
2.1	Olika refererande uttryck	2
2.2	Olika typer av referenter	2
2.3	Syntaktiska och semantiska restriktioner	3
3	Metod	3
4	Resultat	4
5	Slutsats	5
6	Referenser	5
A	Exempel på texter från undersökningen	6
A.1	Datalogitext	6
A.2	Nyhetstext	6
A.3	Skönlitterärtext	6
B	Exempel på konstruerade texter	7
B.1	Refererande uttryck	7
B.1.1	Bestämda substantiv	7
B.1.2	Pronomen	7
B.1.3	Demonstrativa pronomen	7
B.1.4	En-anafor	7
B.2	Svåra referenter	7
B.2.1	Härledda referenter	7
B.2.2	Osammanhängande mängd	7
B.2.3	Allmän referent	7
B.3	Syntaktiska och semantiska restriktioner	7
B.3.1	Numeruskongruens	7
B.3.2	Personkongruens	8
B.3.3	Genuskongruens	8
B.3.4	Verbrestriktioner	8
B.3.5	Syntaktiska restriktioner	8
B.4	Preferenser	8
B.4.1	Nyligen nämnda referenter	8
B.4.2	Grammatisk roll	8
B.4.3	Upprepat omnämnande	8
B.4.4	Parallelism	8
B.4.5	Verbsemantik	8
C	Exempel på algoritmkörning	9

1 Inledning

Samsyftningsresolution (engelska co-reference resolution) är ett problem inom språkteknologi. Forskningsområdet uppstod då man försökte översätta texter med hjälp av datorer. Detta visade sig vara mer komplext än man först trott och har gett upphov till ett antal delproblem. Språkteknologi är det delområde inom AI som behandlar skapandet av naturligt språk och förståelse på mänsklig nivå. Språkteknologi är ett forskningsområde som utvecklas snabbt och det dyker upp nya tillämpningar hela tiden, t.ex. sökmotorer, grammatikkontroll, översättning och talsvarssystem.[1]

Samsyftar gör två eller flera ord som refererar till samma objekt.[2]

(1) Erik pluggar datalogi. Han tycker att det är roligt.

I (1) samsyftar “Erik” och “Han”, de refererar båda till personen Erik. “Erik” och “Han” kallas *refererande uttryck* och det de syftar till kallas *referent*. Det finns flera typer av samsyftning, med anafor menas ett refererande uttryck som syftar till något som nämnts tidigare. “Han” i (1) är en anafor och “Erik” kallas dess korrelat. Ett refererande uttryck som syftar till något som kommer efter kallas katafor.

(2) Innan han gick hemifrån så matade Martin katten.

Ett exempel på en katafor är “han” i (2).

Samsyftningsresolution innebär att man försöker hitta referenten till ett refererande uttryck. Det finns flera olika typer av algoritmer som försöker lösa detta problem och flera olika strategier. Vi kommer i resten av uppsatsen endast prata om algoritmer som tolkar pronomen då detta är det område där det gjorts mest arbete inom samsyftningsresolution. Gemensamt för alla algoritmer är att de sällar bort potentiella referenter som inte passar semantiskt eller syntaktiskt. Det finns också ett antal preferenser som de försöker ta hänsyn till på ett eller annat sätt. Några sådana preferenser är:

- Nyligen nämnda referenter föredras.
- Prioritering enligt grammatisk roll, referenter i subjektposition föredras t.ex. framför de i objekt-position.
- Referenter som upprepat omnämnts i tidigare samtal föredras.
- Verbsemantisk prioritering, vissa verb betonar ett av sina argument vilket leder till att en viss tolkning av senare pronomen föredras.
- Parallelism, vilket föreslår att icke-subjekt pronomen föredrar icke-subjekt referenter.

Det finns ingen algoritm som klarar av att ta hänsyn till alla preferenser och än mindre hantera motsägelser dem emellan.

Lappin och Leass [3] har gjort en algoritm som använder sig av ett enkelt viktsystem för att ta hand om preferenser. För varje pronomen filtrerar algoritmen ut möjliga referenter enligt syntaktiska regler. Sedan väljs en referent enligt en ranking som är baserad på vikter, vikterna är en representation av de preferenser som diskuterats tidigare. Vikterna är framtagna för att passa engelska datalogitexter och kan behöva ändras om man vill applicera algoritmen på andra texter och språk. Ett exempel på hur algoritmen fungerar finns i appendix C.

Hobbs [4] har utvecklat en algoritm som bygger på trädsökning. Algoritmen tar den syntaktiska representationen av meningarna fram till och med den nuvarande och utför en sökning efter en referent. I och med att algoritmen söker i ett syntaxträd så måste en grammatik vara specificerad och input måste bestå av kompletta och syntaktiskt korrekta meningar.

Det finns också algoritmer som kräver inlärnin och använder sig av träningsdata för att beräkna sannolikheten att ett refererande uttryck syftar till en viss referent, ett exempel är Ge, Hale och Charniak [5]. Alla dessa algoritmer kräver input som först har genomgått en komplett analys, en sådan analys består av lemma och ordklassstagning för alla ord. Algoritmernas resultat är starkt beroende av att denna analys är bra, och skapandet av sådana analysverktyg är ett eget forskningsområde.

Det vi skall undersöka i detta arbete är hur Lappin och Leass [3] algoritm klarar sig i jämförelse med människan. I vilka fall är den sämre/bättre? Går den att förbättra och kan vi hitta något sätt att göra det på?

2 Bakgrund

2.1 Olika refererande uttryck

Det finns 5 typer av refererande uttryck som vi vill testa.

- *Obestämda substantiv* introducerar nya referenter, de börjar vanligtvis med en eller ett.

(3) Jag såg en vacker ros idag.

- *Bestämda substantiv* används för att referera till saker läsaren kan identifiera, antingen för att de redan nämnts, för att de är unika eller för att de finns med i läsarens världsbild.

(4) Jag såg en vacker ros idag. Rosen var stor och blå.

I (4) kan referenten till "Rosen" identifieras för att den nämnts tidigare.

- *Pronomen* används vanligtvis för att referera till något som nämnts inte mer än en eller två meningar tidigare. "Den" i (5) är ett exempel på pronomen som ett refererande uttryck.

(5) Jag såg en vacker ros idag. Den var stor och blå.

Man kan också använda pronomen som katafor, i (6) kommer både "han" och "den" före sina referenter.

(6) Innan han köpte den, så frågade Jon om priset på rosen.

- *Demonstrativa pronomen* används ofta för att peka ut något, exempel är "den", "den här" eller "den där". De skiljer ofta på närhet till talaren, vanligtvis fysiskt avstånd men de kan också syfta på annat, t.ex. tid.

(7) Jag köpte en ny mobil igår, den är lik min gamla. Den var bra, men den här har fler bra funktioner.

- *En-anafor* exemplifieras i (8) där "en" refererar till en iPhone.

(8) Jag såg fem iPhone igår. Nu vill jag ha en.

2.2 Olika typer av referenter

Det finns några typer av referenter som försvårar resolutionen.

- Det finns refererande uttryck som går att härleda från en redan introducerad referent (engelska *inferrables*). I (9) kan läsaren förstå att "Bladen" refererar till rosens blad, trots att de inte är explicit introducerade.

(9) Jag såg en vacker ros idag. Bladen var perfekta.

- Dem och de kan referera till uttryck som har introducerats tillsammans som i (10). Men de kan också referera till mängder av referenter som har introducerats i separata fraser, detta kallas *osammanhängande mängd*. "Anja" och "Joel" i (11) är ett exempel på detta.

(10) Anja och Joel älskar sina husdjur. De leker med dem hela tiden.

(11) Anja har en hund och Joel har en katt. De leker med dem hela tiden.

- I (12) har vi ett exempel på en *allmän referent*. "De" syftar inte till de specifika tre katterna utan till katter i allmänhet.

(12) Jag såg tre katter idag. De är det bästa husdjuret.

2.3 Syntaktiska och semantiska restriktioner

Vi vill också undersöka hur väl syntaktiska och semantiska restriktioner hanteras.

- *Numeruskongruens* innebär att refererande uttryck och deras referenter måste stämma överens i antal. I svenskan innebär det att man skiljer på referenter i singular (13) och plural (14).

(13) Jag fick en ros idag. Den är röd.

(14) Jag fick ett dussin rosor idag. De är röda.

- *Personkongruens* illustreras i (15) och (16).

(15) Du och jag har katter. Vi älskar dem.

(16) Du och jag har katter. De älskar dem.

I (16) kan “De” inte referera till “Du och jag”. Kasuskongruens sätter restriktioner på formen av ett pronomen i subjektposition (han, hon), objektposition (honom, henne) och genitivposition (hans, hennes).

- Med *genuskongruens* menas att referentens och det refererande uttryckets kön måste stämma överens. Svenskan skiljer på manligt, kvinnligt och opersonligt.

(17) Anja fick en ros. Hon är vacker.

(18) Anja fick en ros. Den är vacker.

I (17) syftar “Hon” till “Anja”, i (18) syftar “Den” till “en ros”.

- Verb kan sätta restriktioner på sina argument som eliminerar vissa referenter.

(19) Martin stoppade rosen i vasen. Han hade först snittat den.

“den” i (19) har två möjliga referenter, “rosen” och “vasen”. “snittat” kräver dock att dess objekt går att snitta så vi kan utesluta “vasen”. Man kan dock bryta mot dessa restriktioner vid användning av metaforer. För att klara av sådana här restriktioner krävs en djupare kunskap om samtalets deltagare, vilket är anledningen till att många algoritmer inte tar någon större hänsyn till detta.

- När ett refererande uttryck och en potentiell referent är i samma mening finns vissa *syntaktiska restriktioner*. I (20) är “Joel” = “sig” och i (21) är “Joel” $\neq$ “honom”. För dessa finns förenklade regler som tyvärr inte hanterar alla fall då de kan bli ganska komplicerade.

(20) Joel köpte en ny katt åt sig.

(21) Joel köpte en ny katt åt honom.

3 Metod

I undersökningen kommer tio personer med varierande kunskaper inom lingvistik att väljas ut i vår omgivning. Dessa personer kommer först få en kort förklaring om refererande uttryck. Sedan får de instruktionen att markera alla par av referenter och refererande uttryck med penna i givna texter på papper. Dessutom kommer en implementation [6] av Lappin och Leass [3] för anafora resolution söka efter refererande uttryck i samma texter som försökspersonerna fick.

Resultat som är utav intresse är att se hur många refererande uttryck algoritmen respektive försökspersonerna hittar och hur många av dessa som är korrekta. Hur lång tid det tar för algoritmen respektive försökspersonerna att hitta dessa refererande uttryck är inte intressant och kommer inte tas upp i den här rapporten.

Ena delen av texterna som används i undersökningen är slumpmässigt valda och har genrerna skönlitteratur, akademisktext inom datalogi och nyhetsartiklar, exempel på dessa finns i appendix A. Det kommer vara fem texter per genre och varje text omfattar ungefär 15 meningar. Antalet förekommande referenser

i varje text är olika beroende på vilken genre texten tillhör. I genomsnitt förekommer det fem referenser för en nyhetsartikel, åtta för en akademisk (datalogi) text och slutligen tio för en skönlitterär text.

Den andra delen av texter som används i undersökningen är speciellt utformade för att ta upp olika typer av refererande uttryck. Dessa texter är två till tre meningar långa och det kommer vara tre till fyra texter per typ av refererande uttryck. På så sätt kommer det bli enklare att se vilka typer av refererande uttryck som är svåra att upptäcka för algoritmen respektive försökspersonerna.

4 Resultat

	Typ av text	Algoritm	Testpersonerna
Refererande uttryck	Bestämda substantiv	nej	ja
	Pronomen	ja	ja
	Bestämda pronomen	nej	ja
	En-anafor	nej	ja
Svåra referenter	Härledda referenter	nej	ja
	Osammanhängande mängd	delvis	ja
	Allmänna referenter	nej	ja
Syntaktiska och semantiska restriktioner	Nummerkongruens	ja	ja
	Personkongruens	ja	ja
	Genuskongruens	ja	ja
	Syntaktiska restriktioner	delvis	ja
	Verbrestriktioner	nej	ja
Preferenser	Nyligen nämnda referenter	ja	ja
	Grammatisk roll	ja	ja
	Upprepat omnämnande	nej	ja
	Parallelism	nej	ja
	Verbsemantik	nej	ja

Tabell 1: Resultaten för de konstruerade meningarna. Anger om typen av text hanterades korrekt.

Typ av text	Antal pronomen	Algoritmens resultat	Testpersonernas resultat
Datalogitext	40	30 (75%)	40 (100%)
Nyhetsartikel	26	6 (20%)	26 (100%)
Skönlitteratur	53	26 (50%)	53 (100%)

Tabell 2: Resultaten för de längre texterna.

Algoritmens resultat för de längre slumpade texterna, tabell 2, var väldigt dåligt. Vi tror att detta till stor del beror på parsern som använts för att analysera texterna, eftersom vi fick byta ut flera namn och verb i våra konstruerade texter för att få vettiga resultat. Detta visar på hur beroende algoritmen är av en bra parser. På grund av detta är alla våra slutsatser baserade på resultatet från de konstruerade texterna, exempel finns i appendix B. För de konstruerade texterna klarade algoritmen antingen alla eller inga exempel av typen, undantaget osammanhängande mängd och syntaktiska restriktioner där den klarade hälften av meningarna.

Bland de olika typerna av refererande uttryck hanterar algoritmen bara pronomen, vilket är vad vi förväntade oss då det är en algoritm för samsyftningsresolution av pronomen. Av samma anledning klarar den inte av härledda eller allmänna referenter.

Hos de syntaktiska och semantiska restriktionerna klarar algoritmen de tre kategorierna nummer-, person- och genuskongruens, men får problem med svårare syntaktiska restriktioner och hanterar inte verbrestriktioner alls. Detta tyder på att verbrestriktioner inte implementerats i algoritmen och bara grundläggande stöd för syntaktiska restriktioner finns med. Anledningen till detta är antagligen skillnaden i svårighet att implementera dessa restriktioner. För att hantera verbrestriktioner krävs mer information om verbet än den som fås i input.

Algoritmen tar hänsyn till att nyligen introducerade referenter och vissa grammatiska roller föredras. Den verkar dock preferera dessa för stark i vissa situationer, speciellt i förhållande till referenter som upprepat omnämnts. Parallelism och verbsemantiska preferenser tar algoritmen ingen hänsyn till.

Resultaten från testet utförd av människor visade att inga felaktiga samsyftningar förekom, däremot upptäcktes inte alla samsyftningar i de längre texterna. Detta kan bero på att testpersonen tolkar uppgiften att hitta samsyftningar, eller refererande uttryck, olika. Vissa testpersoner markerade även refererande uttryck vars referent inte var nämnd i texten, detta är ett typiskt exempel på något som kräver en djupare förståelse av sammanhanget än vad dagens algoritmer kan klara av.

Den typ av text som var enklast för testpersonerna att hitta refererande uttryck i var den skönlitterära. Anledningen till detta är förmodligen att skönlitterära texter är väldigt beskrivande och det förekommer inte en form av kommunikation mellan författaren och läsaren. I text från en lärobok försöker författaren nå ut till läsaren medan i en text från en skönlitterär bok existerar inte läsaren. De enda typer av text som gav upphov en viss tveksamhet hos våra testpersoner var exempel på parallelism och upprepat omnämnande. Dessa uppfattades lätt som tvetydiga.

5 Slutsats

Algoritmen klarar det mesta som man förväntar sig om man är insatt i hur den är upplagd, men om man tittar på undersökningen så inser man hur mycket algoritmen saknar stöd för. Att lägga till detta stöd är långt ifrån trivialt och den här typen av enkel viktbaserad algoritm lämpar sig troligen inte för det heller. En lärande algoritm krävs antagligen för att hantera sådant som parallelism, verbpåverkan och svårare syntaktiska restriktioner.

Problemet att hitta refererande uttryck är för oss människor trivialt. Det är så självklart att vi många gånger inte ens tänker på att något är ett refererande uttryck. För datorer är det ett helt annat problem, utöver några få tydliga enkla regler så finns en myriad specialfall, undantag och tolkningar som kräver en större förståelse av sammanhang. Så det dröjer innan algoritmerna kommer i närheten av det vi människor klarar av på området.

6 Referenser

- [1] http://en.wikipedia.org/wiki/Computational_linguistics
- [2] D. Jurafsky and J. Martin. Speech and Language Processing, chapter 18.1. Prentice-Hall, 2000
- [3] S. Lappin and H. Leass. An Algorithm for Pronominal Anaphora Resolution. <http://acl.ldc.upenn.edu/J/J94/J94-4002.pdf>
- [4] J. Hobbs. Resolving pronoun references. <http://www.isi.edu/~hobbs/ResolvingPronounReferences.pdf>
- [5] N. Ge, J. Hale and E. Charniak. A Statistical Approach to Anaphora Resolution. <http://sca2002.cs.brown.edu/research/pubs/pdfs/1998/Ge-1998-SAA.pdf>
- [6] JavaRAP. <http://wing.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html>
- [7] C. Kennedy and B. Boguraev. Anaphora for everyone: Pronominal anaphora resolution without a parser. http://www.research.ibm.com/talent/documents/bran_anaphora_coling96.pdf

A Exempel på texter från undersökningen

A.1 Datalogitext

A C++ program is a collection of commands, which tell the computer to do “something”. This collection of commands is usually called C++ source code, source code or just code. Commands are either “functions” or “keywords”. Keywords are a basic building block of the language, while functions are, in fact, usually written in terms of simpler functions—you’ll see this in our very first program, below. (Confused? Think of it a bit like an outline for a book; the outline might show every chapter in the book; each chapter might have its own outline, composed of sections. Each section might have its own outline, or it might have all of the details written up.) Thankfully, C++ provides a great many common functions and keywords that you can use. But how does a program actually start? Every program in C++ has one function, always named `main`, that is always called when your program first executes. From `main`, you can also call other functions whether they are written by us or, as mentioned earlier, provided by the compiler. So how do you get access to those prewritten functions? To access those standard functions that comes with the compiler, you include a header with the `#include` directive. What this does is effectively take everything in the header and paste it into your program.

A.2 Nyhetstext

Defence Secretary Robert Gates had given Congress notice that the ban would end unless it objected and that deadline passed at midnight on Wednesday. Navy Secretary Ray Mabus said in a statement: There are extremely capable women in the navy who have the talent and desire to succeed in the submarine force. “We literally could not run the navy without women today. The first women submariners are likely to be aboard ballistic and guided-missile submarines as they have more living space and will not require alterations. However, there is a 15-month submarine officer training course. Chief of Naval Operations, Adm Gary Roughead, said: “It would be foolish to not take the great talent, the great confidence and intellect of the young women who serve in our navy today and bring that into our submarine force. The US navy has also announced another key change to service requirements. As of 31 December, smoking will no longer be allowed on submarines. There are estimated to be more than 5,000 submariner smokers but they will be expected to fall in line with the general ban on indoor smoking the military imposed in 1993.

A.3 Skönlitterärtext

There were once a man and a woman who had long, in vain, wished for a child. At length it appeared that God was about to grant their desire. These people had a little window at the back of their house from which a splendid garden could be seen, which was full of the most beautiful flowers and herbs. It was, however, surrounded by a high wall, and no one dared to go into it because it belonged to an enchantress, who had great power and was dreaded by all the world. One day the woman was standing by this window and looking down into the garden, when she saw a bed which was planted with the most beautiful rampion, and it looked so fresh and green that she longed for it. She quite pined away, and began to look pale and miserable.

B Exempel på konstruerade texter

Algoritmens resultat skrivs på formen:

(meningsnummer,ordnummer) referent $\leftarrow$ (meningsnummer,ordnummer) refererande uttryck
eller “null” om den inte hittade något refererande uttryck.

B.1 Refererande uttryck

B.1.1 Bestämda substantiv

I saw an Acura Integra today. The Integra was white and needed to be washed.
null

B.1.2 Pronomen

I saw an Acura Integra today. It was white and needed to be washed.
(0,2) an Acura Integra $\leftarrow$ (1,0) It
Before he bought it, Bob checked over the Integra very carefully.
(0,9) Bob $\leftarrow$ (0,1) he, (0,9) Integra $\leftarrow$ (0,3) it

B.1.3 Demonstrativa pronomen

I bought an Integra yesterday. It's similar to the one I bought five years ago. That one was really nice, but I like this one even better.
(0,2) an Integra $\leftarrow$ (1,0) It

B.1.4 En-anafor

I saw no less than 6 Acura Integras today. Now I want one.
null

B.2 Svåra referenter

B.2.1 Härledda referenter

I almost bought an Acura Integra today, but a door had a dent and the engine seemed noisy.
null

B.2.2 Osammanhängande mängd

Bob has an Acura, and Mary has a Mazda. They drive them all the time.
(0,0) Bob has an Acura, and Mary $\leftarrow$ (1,0) They, (0,0) Bob has an Acura, and Mary $\leftarrow$ (1,2) them

B.2.3 Allmän referent

I saw no less than 6 Acura Integras today. They are the coolest cars.
NULL $\leftarrow$ (1,0) They

B.3 Syntaktiska och semantiska restriktioner

B.3.1 Numeruskongruens

Bob has a new Acura. It is red.
(0,2) a new Acura $\leftarrow$ (1,0) It
Bob has three roses. They are red.
(0,2) three roses $\leftarrow$ (1,0) They

B.3.2 Personkongruens

Bob and Mary have Acuras. They love cars.
 (0,0) Bob and Mary $\leftarrow$ (1,0) They

B.3.3 Genuskongruens

Bob has an Acura. He is attractive.
 (0,0) Bob $\leftarrow$ (1,0) He
 Bob has an Acura. It is attractive.
 (0,2) an Acura $\leftarrow$ (1,0) It

B.3.4 Verbrestriktioner

Bob parked his car in the garage. It is very messy, with old bike and car parts lying around.
 (0,0) Bob $\leftarrow$ (0,2) his, (0,2) his car $\leftarrow$ (1,0) It

B.3.5 Syntaktiska restriktioner

Bob bought himself a new Acura.
 (0,0) Bob $\leftarrow$ (0,2) himself
 Bob bought him a new Acura.
 NULL $\leftarrow$ (0,2) him
 Bob wanted a new car. Bill bought him a new Acura.
 (0,0) Bob $\leftarrow$ (1,2) him
 Bob wanted a new car. He bought him a new Acura.
 (0,0) Bob $\leftarrow$ (1,0) He, (0,0) Bob $\leftarrow$ (1,2) him

B.4 Preferenser**B.4.1 Nyligen nämnda referenter**

Bob has an Integra. Bill has a Legend. Mary likes to drive it.
 (1,2) a Legend $\leftarrow$ (2,4) it

B.4.2 Grammatisk roll

Bob went to the Acura dealership with Bill. He bought an Integra.
 (0,0) Bob $\leftarrow$ (1,0) He
 Bill went to the Acura dealership with Bob. He bought an Integra.
 (0,0) Bill $\leftarrow$ (1,0) He

B.4.3 Upprepat omnämmande

Bob needed a car to get to his new job. He decided that he wanted something sporty. Bill went to the Acura dealership with him. He bought an Integra.
 (0,0) Bob $\leftarrow$ (0,7) his, (0,0) Bob $\leftarrow$ (1,0) He, (0,0) Bob $\leftarrow$ (1,3) he, (2,0) Bill $\leftarrow$ (2,7) him, (2,0) Bill $\leftarrow$ (3,0) He

B.4.4 Parallelism

Mary went with Sue to the Acura dealership. Sally went with her to the Mazda dealership.
 (1,0) Sally $\leftarrow$ (1,3) her

B.4.5 Verbsemantik

John telephoned Bill. He lost the pamphlet on Acuras.
 (0,2) Bill $\leftarrow$ (1,0) He
 John criticized Bill. He lost the pamphlet on Acuras.
 (0,2) Bill $\leftarrow$ (1,0) He

C Exempel på algorithmkörning

I följande exempel går vi igenom meningarna: “John såg en vacker Acura Integra hos bilhandlaren. Han köpte den.” ungefär så som Lappin och Leass algoritim [3] gör det.

- Gå igenom första meningen och samla potentiella referenter och beräkna deras initiala vikter. Vi hittar tre potentiella referenter “John”, “Integra” och “bilhandlaren”. Viktberäkningen finns i följande tabell.

	Nyligen	Subj	Exist	Obj	Indir Obj	Icke adv	Huvud subst	Summa
John	100	80				50	80	310
Integra	100			50		50	80	280
bilhandlaren	100					50	80	230

Då det inte finns något pronomen i meningen delas värdena med två innan vi går vidare till nästa mening.

John	155
Integra	140
bilhandlaren	115

- Gå igenom andra meningen, då det inte finns några nya referenter letar vi pronomen. Vi hittar två stycken “Han” och “den”. Eftersom “Han” specificerar manligt kön så är “John” den enda möjliga referenten. Värdet för “John” uppdateras med vikterna för “Han” (Nyl. 100 + Subj 80 + Icke adv 50 + Huvud subs 80 = 310).

John	465
Integra	140
bilhandlaren	115

“den” är kompatibel med både “Integra” och “bilhandlaren” så slutgiltiga vikter måste beräknas. Ingen av tilldelningarna skulle resultera i katafora men “Integra” och “den” är båda i objektposition så vikten för “Integra” ökas med 35. “Integra” väljs som referent då 175 är mer än 115, och dess värde uppdateras med vikterna för “den” (Nyl. 100 + Obj 50 + Icke adv 50 + Huvud subs 80 = 280).

John	465
Integra	420
bilhandlaren	115

Då var vi färdiga och har fått resultatet:

(0,0) John <- (1,0) Han

(0,5) Integra <- (1,2) den

