

TIAGO SANTOS DE LIMA



**ANÁLISE SEMÂNTICA DO OBJETO NULO EM
PORTUGUÊS DO BRASIL**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Dr. Michel Gagnon

Co-orientador: Dr. José Borges Neto

CURITIBA

2002

TIAGO SANTOS DE LIMA

**ANÁLISE SEMÂNTICA DO OBJETO NULO EM
PORTUGUÊS DO BRASIL**

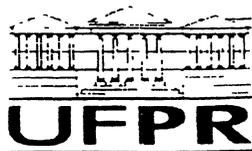
Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Dr. Michel Gagnon

Co-orientador: Dr. José Borges Neto

CURITIBA

2002



Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, do aluno *Tiago Santos de Lima*, avaliamos o trabalho intitulado. "*Análise Semântica do Objeto Nulo em Português do Brasil*", cuja defesa foi realizada no dia 11 de dezembro de 2002, às quatorze horas, no anfiteatro B do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação do candidato.

Curitiba, 11 de dezembro de 2002.

Prof. Dr. Michel Gagnon
DINF/UFPR – Orientador

Prof. Dra. Renata Vieira
UNISINOS - Membro Externo

Prof. Dr. José Borges Neto
UFPR - Co-orientador

Prof. Dr. Jair Donadelli Júnior
DINF/UFPR



Agradecimentos

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Programa de Pós-graduação em Informática da Universidade Federal do Paraná pelo apoio financeiro e oportunidade a mim oferecidos. Agradeço também aos meus amigos, à minha família e aos meus pais cujo apoio e compreensão foram determinantes na realização deste trabalho. Além deles, minha sincera gratidão aos professores Michel Gagnon (orientador), José Boiges Neto (co-orientador) e Alexandre I. Direne (ex-professor e amigo).

*Última flor do Lácio, inculta e bela.
És, a um tempo, esplendor e sepultura:
Ouro nativo, que na ganga impura
A bruta mina entre os cascalhos vela...*

(Olavo Bilac)

Sumário

LISTA DE FIGURAS	vii
RESUMO	viii
ABSTRACT	ix
1 Introdução	1
2 Definição do problema	6
2.1 Fenômenos fóricos	6
2.2 O objeto nulo	7
2.2.1 Representação do vazio	7
2.2.2 Interferência da transitividade do verbo	8
2.2.3 Identidade entre objetos	9
2.2.4 Papel dos prossintagmas	11
2.3 Considerações finais	12
3 Quadro teórico	13
3.1 Sintaxe	13
3.2 Semântica	17
3.2.1 Teoria de representação de discurso	19
3.2.2 O cálculo lambda	27
3.2.3 λ -DRS	28
3.3 Resolução de fenômenos fóricos	32
3.3.1 Gênero e número	33
3.3.2 Contra-índice	33
3.3.3 Consistência semântica	33
3.3.4 Paralelismo	35
3.3.5 Saliência	36
3.3.6 α -DRT	41
3.3.7 Juntando tudo	44
3.4 Considerações finais	48
4 Nova proposta para o objeto nulo	50
4.1 Condições para apagamento do objeto	51
4.2 Um algoritmo para resolução do fenômeno	54
4.2.1 Novo modelo de conhecimento de mundo	55
4.2.2 Zero-anáfora	58
4.2.3 Elipse de SV^+	61

4.3	Limitações do algoritmo	80
4.3.1	Regras de aceitação	80
4.3.2	Objetos nulos sentenciais	82
4.4	Implementação	83
5	Conclusão	86
	REFERÊNCIAS BIBLIOGRÁFICAS	88
	ÍNDICE REMISSIVO	91

Lista de Figuras

3.1	Ilustração geométrica do conceito de acessibilidade na DRT.	25
3.2	Árvore sintática: João ama Maria.	31
3.3	Ontologia de Saiz-Neoda e Palomar (O_n e O_r)	34
3.4	Árvore sintática: Ele ama maria.	43
4.1	Nova ontologia (O_n , O_r e O_a)	57
4.2	Árvore sintática: Maria vendeu. (zero-anáfora)	59
4.3	Árvore sintática: Ana comprou um bolo. (eclipse de SV)	69
4.4	Árvore sintática: Maria vendeu. (transitivo)	70
4.5	Árvore sintática: Maria vendeu. (intransitivo)	70
4.6	Árvore sintática: João odeia comer arroz.	74
4.7	Árvore sintática: Maria também.	74
4.8	Árvore sintática: Maria, feijão.	77

Resumo

Este trabalho reúne um conjunto de procedimentos capazes de mapear a representação semântica de discursos escritos em linguagem natural (mais precisamente, no Português do Brasil) em um formalismo lógico. O enfoque principal é dado a discursos contendo um fenômeno chamado objeto nulo. Este fenômeno é definido como sendo a referência a um objeto extra-lingüístico na posição de complemento verbal que, todavia, não aparece explícito no discurso (ex.: João comeu o bolo. Pedro não comeu \emptyset). Aquilo que Pedro não comeu (simbolizado por \emptyset) é o objeto nulo. Solucionar este fenômeno implica em determinar qual é o elemento apagado e construir sua representação semântica, combinando-a com a do resto do discurso. Com este objetivo, proponho um procedimento mais geral, que trata do apagamento de alguns outros elementos do sintagma verbal da sentença. Dois fenômenos são distingüidos: a zero-anáfora e a elipse de *SV*. O primeiro deles é resolvido da mesma forma que os demais fenômenos fóricos da linguagem, enquanto que o segundo emprega um procedimento prévio de análise dos sintagmas verbais e copia os elementos elididos no nível sintático. Além disso, uma revisão de várias teorias é feita. A presente abordagem integra os seguintes itens: (1) o procedimento descrito há pouco; (2) uma gramática com estruturas de traços; (3) um formalismo lógico dinâmico chamado de DRT; (4) um algoritmo que soluciona correferências fóricas e (5) uma ontologia de objetos do mundo.

Abstract

This work assembles a set of procedures capable of mapping the semantic representation of discourses written in natural language (more precisely, in Brazilian Portuguese) into a logic formalism. The main focus is given to discourses which contain a phenomenon called the null object. This phenomenon is defined as a reference to an extra-linguistic object filling in the position of a verb complement which, however, does not appear explicitly in the discourse (*e.g.*, *João comeu o bolo. Pedro não comeu \emptyset* – João ate the cake. Pedro didn't eat \emptyset). The thing that Pedro did not eat (symbolized by \emptyset) is the null object. To solve this phenomenon implies determining which is the hidden element and to construct its semantic representation, linking it together with the rest of the discourse. With this purpose, I propose a more general procedure, which treats the deletion of certain elements of the verb phrase. Two phenomena are distinguished: zero-anaphora and *VP* ellipsis. The former is solved in the same way as in the anaphora phenomena of the language, while the latter utilizes a pre-analysis procedure of the verb phrases and copies the hidden elements on the syntactic level. Moreover, a revision of several theories is done. The current approach integrates the following items: (1) the procedure described above; (2) a grammar with phrase structures; (3) a dynamic logic formalism called DRT; (4) an algorithm that solves anaphoric references and (5) an ontology of world objects.

Capítulo 1

Introdução

Encarada como um aspecto importante do comportamento humano, a linguagem natural (ou língua natural) serve como meio de comunicação e transmissão de conhecimento. É através desta que as pessoas, durante toda sua vida, aprendem sobre o mundo exterior, sobre si mesmos e expressam seus sentimentos, desejos e opiniões. O estudo deste mecanismo leva ao entendimento do próprio ser humano. Logo, não é a toa que as pesquisas nesta área abrangem várias disciplinas, cada uma com seu próprio conjunto de problemas e métodos de resolução.

Na intersecção entre a Linguística e a Ciência da Computação, surge a Linguística Computacional. Objetivando o desenvolvimento de teorias computacionais para a linguagem natural, usa noções de algoritmos, estruturas de dados, modelos formais de representação e técnicas de inteligência artificial para compreender a maneira pela qual as estruturas lingüísticas são identificadas. Em última instância, busca por uma melhor compreensão dos mecanismos lingüísticos utilizados pelo ser humano, aproveitando a capacidade dos computadores de implementar e ajudar no teste e avaliação das teorias desenvolvidas. Além disso, visa também uma evolução na própria utilização dos computadores, tornando a forma de interação entre o homem e a máquina mais amigável e eficiente. Finalmente, procura aproximar o próprio computador à capacidade humana de realização de tarefas cada vez mais complexas.

O alcance de tais objetivos obriga a solução de inúmeros problemas. Em sua obra, Allen ([2]) observa várias questões a serem analisadas: O que as palavras são? Como combiná-las para formar sentenças? O que significam? Como seus significados contribuem para o significado de uma sentença? Aliado a isto, se um sistema computacional espera comportar-se de maneira semelhante a um ser humano, deve levar em conta conhecimento e poder de raciocínio. O autor então classifica o conhecimento necessário para solução destes problemas em vários níveis:

- Fonético e fonológico: refere-se ao relacionamento das palavras aos seus respectivos sons.

- Morfológico: refere-se a como as palavras são construídas através de radicais e sufixos.
- Sintático: refere-se a regras estruturais de como palavras podem ser combinadas para formar sentenças.
- Semântico: refere-se ao significado das palavras e como seus significados podem ser combinados para capturar o significado de uma sentença.
- Pragmático: refere-se a como as palavras são utilizadas em diferentes situações e como seu uso afeta a interpretação de uma sentença.
- Discurso: refere-se a como as sentenças colocadas em conjunto afetam o significado umas das outras.
- Conhecimento de Mundo: refere-se ao conhecimento da estrutura e do funcionamento do mundo ou da situação em que está inserido o discurso.

Estas definições são imprecisas e polêmicas. A própria classificação dos problemas nos níveis de conhecimento é difícil. Frequentemente, encontram-se questões que não podem ser respondidas sem a utilização de conhecimentos atribuídos a vários destes níveis. Um deles, a *resolução de fenômenos fóricos*, é o objeto de estudo deste trabalho. Informações provenientes dos níveis sintático, semântico, pragmático, do discurso e do conhecimento de mundo precisam ser analisadas em conjunto para a produção de uma resposta aceitável.

Um fenômeno fórico é uma referência não explícita a um objeto. O exemplo (1.1) ilustra uma referência fórica do pronome oblíquo *o* ao bolo, referido pelo sintagma *o bolo*.

(1.1) João ofereceu o bolo, mas Pedro não **o** quis.

Os pronomes são exemplos de fenômenos fóricos frequentemente encontrados em discursos em linguagem natural. Existem, porém, referências não explícitas oriundas da ausência de elementos em um discurso.

(1.2) João comeu o bolo. Pedro não comeu \emptyset .

Simbolizada por \emptyset , a ausência do complemento do verbo¹ *comeu* também é interpretada como uma referência ao bolo. Este fenômeno é chamado de *zero-anáfora*. Quando uma zero-anáfora ocorre na posição de objeto do verbo, ela também recebe o nome de *objeto nulo*.

¹Esta sentença é, na realidade, ambígua. O verbo *comer* pode ser intransitivo, significando que Pedro deixou de realizar uma refeição, por exemplo. Porém, para esta discussão, assumo apenas o caso em que ele é transitivo, que aliás, considero mais natural. Este assunto será mais amplamente discutido na seção 2.2.2.

Segundo os estudos de Cyrino ([7, 8]), o Português do Brasil (PB) emprega o objeto nulo desde a chegada da linguagem a este país. Nesta ocasião, a utilização deste era mais restrita que a atual, mas com o passar do tempo, o PB se distanciou do Português europeu (PE) na medida em que diminuía as restrições ao emprego do objeto nulo. Raposo, em [34], oferece uma série de exemplos que ilustram muito bem isto.

(1.3) A Joana viu \emptyset na TV ontem.

(1.4) O rapaz que trouxe \emptyset agora mesmo da pastelaria era o teu afilhado.

(1.5) Que a IBM venda \emptyset a particulares surpreende-me.

(1.6) O pirata partiu para as Caraíbas depois de ter guardado \emptyset cuidadosamente no cofre.

Nestes exemplos, existem referências feitas a elementos fora do escopo do discurso. Por exemplo, em (1.4), existe algo que o rapaz trouxe da pastelaria, mas a referência a este elemento, simbolizada por \emptyset , não está explícita. É, portanto, subentendida.

Segundo Raposo, apenas o exemplo (1.3) é aceito no PE. No PB, no entanto, todas as ocorrências acima são aceitas. Observe que estas diferenças aumentam a importância do estudo deste fenômeno, pois trata-se de algo que, provavelmente, não ocorre da mesma forma em nenhuma outra linguagem.

O presente trabalho propõe um tratamento computacional para o objeto nulo no Português do Brasil. Dado um discurso contendo tal fenômeno, o procedimento descrito aqui retorna sua representação semântica em um formalismo lógico. Este processo engloba vários passos distintos. Primeiramente, uma seqüência de palavras é reconhecida pelas regras de boa formação da linguagem. O significado individual de cada palavra é capturado e, juntamente com as informações sintáticas, é passado para o passo seguinte. O segundo passo utiliza as informações recebidas na construção do significado completo do discurso em uma linguagem de representação lógica. Ao mesmo tempo, um algoritmo de saliência, filtros sintáticos, semânticos e informações do contexto são utilizados para resolver as ocorrências fóricas.

Todo este processo revela-se bastante complexo e encerra vários desafios. O primeiro deles advém da própria natureza do fenômeno. Como visto no exemplo (1.2), não há um elemento explicitando a relação de referência. Bem como nenhuma pista, como as comumente dadas por pronomes, sobre o gênero, número ou tipo de objeto ao qual o pronome se refere. Aliás, no caso específico do exemplo citado, não há nem mesmo a certeza da existência de um objeto nulo, pois pode-se interpretar o verbo como intransitivo também.

Explorando mais profundamente o problema, outras perguntas surgem. Em (1.2), Pedro comeu o mesmo bolo que João. Mas considerando o mesmo exemplo ligeiramente modificado para

(1.7) João comeu um bolo. Pedro não comeu \emptyset .

é novamente possível afirmar que Pedro comeu um bolo, mas não o mesmo bolo que João comeu. Ou seja, supondo que o procedimento seja capaz de descobrir quais elementos estão sub-entendidos, a representação semântica precisa prever casos em que os objetos referenciados podem não ser os mesmos.

A tentativa de resolver casos como o apresentado em (1.7) levou a um estudo mais abrangente de formas de apagamento que, mais tarde, revelaram possuir relação entre si. O estudo de casos como

- (1.8) (a) João comeu um bolo. Pedro também \emptyset .
(b) João comeu um bolo. Pedro não \emptyset .
(c) João comeu um bolo. Pedro, \emptyset uma torta.

revelou, além das regras gramaticais que permitem o seu aparecimento em um discurso, o procedimento para encontro do referente para os elementos apagados. Este baseia-se em um procedimento de cópia dos constituintes de sentenças anteriores.

A correta interpretação de fenômenos fóricos constitui peça fundamental na Linguística Computacional. Profundamente ligada à coesão do discurso, sua interpretação semântica figura como um campo de pesquisa absolutamente aberto e relevante em vários contextos. Dentre os mais comuns, podem ser citados a procura de documentos em bases de dados textuais, extração de informações de mensagens escritas ou artigos, tradutores automáticos de textos, sumarização, sistemas de resposta a questões, atendimento automatizado de clientes, sistemas tutoriais inteligentes, controle de máquinas via voz e sistemas cooperativos para resolução de problemas.

O segundo capítulo deste texto define precisamente o problema a ser abordado pela pesquisa aqui descrita. O conceito de fenômeno fórico é apresentado de maneira mais formal e o termo *objeto nulo* também é definido. O segundo capítulo também descreve os vários problemas encontrados na tentativa de solução do fenômeno, dando uma pequena idéia de como cada um deles poderia ser solucionado.

Adiante, no capítulo 3, são apresentadas as teorias lingüístico-computacionais utilizadas neste trabalho. Sua disposição segue, de forma não muito rígida, os níveis de conhecimento descritos por Allen ([2]), apresentados anteriormente. Primeiramente, discorro sobre o reconhecimento sintático, em seguida sobre a representação semântica e por último sobre algumas técnicas de resolução de fenômenos fóricos.

Uma proposta para solução de fenômenos como os apresentados nos exemplos (1.2), (1.7) e (1.8) é descrita no capítulo 4. Primeiramente, é feito um destrinchamento dos exemplos, classificando-os entre tipos distintos e, nos casos em que foi possível, apresento

uma solução computacional. Também nesse capítulo, apresento algumas limitações do procedimento proposto através de alguns exemplos para os quais não são obtidas respostas completamente aceitáveis, apontando novos desafios a serem enfrentados e perspectivas para trabalhos futuros.

O último capítulo desta dissertação conclui o trabalho. O capítulo também destaca as contribuições dadas e a relevância da pesquisa realizada no âmbito da Linguística Computacional.

Capítulo 2

Definição do problema

O objeto nulo, foco principal desta pesquisa, é um complemento verbal que precisa ser resgatado do contexto. Cyrino, em [9], afirma que este figura como um fenômeno fórico com características especiais, pois o elemento de referência não está presente. Este capítulo define a noção de fenômeno fórico, a de objeto nulo e, logo em seguida, descreve alguns problemas a serem enfrentados na tentativa de interpretar discursos contendo tais fenômenos.

2.1 Fenômenos fóricos

Pode-se definir um fenômeno fórico como uma referência indireta a um elemento do contexto atual do discurso. Seguindo as definições de referência dadas por Yule em [41], e por Larson e Segal em [21], este fenômeno pode ser entendido como a utilização de elementos lingüísticos (*elementos fóricos*) na identificação de objetos extra-lingüísticos sem a utilização explícita dos seus respectivos nomes. Pronomes pessoais como 'ele' e 'ela' ou demonstrativos como 'isto' e 'aquilo' são exemplos encontrados com bastante freqüência.

Comumente, elementos fóricos referenciam objetos explicitamente citados no discurso, perfazendo uma relação de *correferência*. Esta relação é mais rigorosamente definida por Greimas e Courtés em [14]: correferência é a relação que dois signos lingüísticos (idênticos ou diferentes) mantém entre si, quando situados em dois lugares (contíguos ou distanciados) da cadeia falada ou escrita, remetem a um mesmo objeto extra-lingüístico.

Um exemplo simples dos conceitos apresentados pode ser observado em (2.1) abaixo.

(2.1) Pedro ama Maria, Ela, é bonita.

O pronome Ela é um elemento fórico que possui uma relação de correferência com o nome Maria. Ambos remetem a um mesmo objeto extra-lingüístico: uma pessoa chamada Maria. Quando o elemento correferente a um elemento fórico se encontra antes deste no discurso,

como é o caso de Maria, ele também pode ser chamado de *antecedente*. Doravante, uma relação entre dois termos lingüísticos será indicada por índices iguais (*i*, no exemplo acima) subscritos às palavras.

2.2 O objeto nulo

() Objeto nulo é um elemento não explícito em posição de objeto de um verbo, ou seja, um complemento verbal que não aparece explicitamente no discurso. Para ilustrar um caso típico, o exemplo (1.2) é repetido aqui.

(2.2) João comeu o bolo_i. Pedro não comeu \emptyset _i.

Pode-se dizer que o objeto do verbo comeu na segunda sentença está subentendido ou que ele é nulo. Cyrino afirma, em [9], que tanto no Português do Brasil como nas outras linguagens que o suportam, o objeto nulo se apresenta como um fenômeno fórico. Desta forma, toda vez que um objeto nulo é identificado, torna-se necessário buscar o elemento do contexto atual do discurso ao qual ele se refere.

O objeto nulo pode aparecer em várias posições na sentença. Todavia, ele certamente não é irrestrito. Existem construções nas quais a ausência de complemento verbal não é um objeto nulo. Ou seja, não existe nenhuma referência subentendida.

(2.3) *João comeu feijão_i. Maria comeu \emptyset _i.¹

Fica claro, portanto, que algumas regras precisam ser estabelecidas no concernente ao tratamento do fenômeno. As seções seguintes analisam alguns exemplos de apagamento do complemento verbal. O objetivo é de evidenciar algumas destas regras e de apontar problemas no tratamento computacional deste fenômeno.

2.2.1 Representação do vazio

Sabe-se que é preciso extrair dos itens léxicos os componentes da representação semântica do discurso. A união de todos estes componentes deve formar a representação completa do discurso. Como o reconhecimento sintático da linguagem é feito através de uma gramática, seria natural representar um constituinte que não se faz presente, como um objeto nulo por exemplo, através de uma "transição vazia". Ou seja, uma regra gramatical que não produz elemento algum, mas que contribui na construção da representação semântica. Em outras palavras, o reconhecedor sintático, mesmo não encontrando um

¹O símbolo '*' denota uma sentença agramatical ou sem interpretação semântica possível. Embora este exemplo não seja agramatical e possa ser interpretado no caso do verbo comeu ser intransitivo, o símbolo '*' deve-se ao fato da interpretação do exemplo com \emptyset não ser possível.

constituente apropriado na posição de objeto, simularia sua presença com uma “palavra vazia” (um ϵ) atribuindo a esta uma representação capaz de ser resolvida como um fenômeno fórico comum. Isto funciona bem em casos como o do exemplo (2.4) abaixo.

(2.4) Pedro vendeu um carro, João ϵ , comprou.

O ϵ é produzido pela gramática no lugar do objeto do verbo e pode, por exemplo, possuir a mesma representação do pronome oblíquo ‘o’. Assim, este exemplo seria resolvido como se a segunda sentença fosse João o comprou.

Porém, este tratamento gera um novo problema: como impedir que a gramática produza ϵ em lugares indevidos? O exemplo (2.5) ilustra um caso como este.

(2.5) *Pedro chamou José. João ϵ , comprou.

Como o verbo comprou pode ser tanto transitivo como intransitivo, o algoritmo encontraria mais de uma interpretação para a segunda sentença do exemplo. A interpretação intransitiva pode ser aceita, mas a interpretação na qual a gramática produz ϵ está incorreta. Esta é equivalente a João o comprou. O pronome na posição de complemento do verbo seria correferente de José². Supondo que seja possível impedir esta interpretação, ainda existe uma outra, na qual o ϵ simularia o pronome reflexivo ‘se’. Agora, o correferente de ϵ é João. Nenhuma destas interpretações está correta, donde surge a necessidade de uma abordagem diferente.

É preciso saber quais são os casos em que a gramática pode ou não produzir ϵ . Adiantando um pouco a solução apresentada no capítulo 4, os candidatos a correferente de ϵ no exemplo (2.5) denotam objetos animados, enquanto que um dos candidatos de (2.4) não é animado. Não é coincidência o fato deste candidato ser o correferente correto.

2.2.2 Interferência da transitividade do verbo

Assim como o verbo comprar visto há pouco, vários verbos possuem interpretações transitivas e intransitivas. Por esta razão, nem toda ausência de objeto pode ser interpretada como objeto nulo. O exemplo (1.2), rerepresentado aqui, ilustra um caso como este.

(2.6) João comeu o bolo. Pedro não comeu.

A interpretação mais natural considera o verbo comeu como transitivo direto. Neste caso, entende-se que Pedro teria deixado de comer o bolo. Porém, considerando o verbo em questão como intransitivo, uma outra interpretação é possível. Neste caso, a segunda sentença significaria que Pedro deixou de fazer uma refeição, por exemplo.

Outro exemplo interessante surge com um verbo no infinitivo.

²Na realidade, existe uma interpretação na qual o pronome pode ser correferente de uma entidade fora do escopo do discurso. Neste trabalho, contudo, não trato este tipo de fenômeno.

(2.7) João arquivou o artigo sem ler.

Novamente, é possível duas interpretações. Numa delas, ler possui um objeto, artigo. Na outra, ler é intransitivo.

Também existem casos em que a interpretação transitiva do verbo é proibida.

(2.8) João foi ao restaurante com Maria. Ela não comeu.

Na maioria dos casos em que este tipo de ambigüidade ocorre, existe uma interpretação mais aceita do que outra. Porém, esta escolha não se faz utilizando informações presentes no discurso. Utiliza-se informações acerca do contexto no qual o discurso é proferido. Dados sobre quem o profere, quando, e de que maneira, influenciam de forma decisiva. Logo, a representação computacional destes fatos é demasiadamente complexa. Além disso, a simples representação não é suficiente. Aproveitar estas informações requer um mecanismo de inferência poderoso. Portanto, este problema não é resolvido pela presente proposta. Todavia, o procedimento de interpretação criado é capaz de retornar todas as representações possíveis, assumindo que uma abordagem mais profunda do problema possa ser dada futuramente.

2.2.3 Identidade entre objetos

A existência de uma relação de correferência entre dois signos lingüísticos pressupõe a existência de um único objeto extra-lingüístico ao qual os dois signos se referem. Contudo, a representação formal de cada um dos signos lingüísticos cria, em princípio, uma entidade para cada um e em seguida, identifica das duas entidades. Ou seja, as duas entidades são interpretadas como sendo a mesma. Por exemplo, assumindo apenas a interpretação transitiva do verbo comeu, a representações em lógica de primeira ordem de

(2.9) Maria comprou uma maçã, . Pedro não comeu \emptyset ,

poderia ser dada por

$$\exists x[maca(x) \wedge comprou(maria, x) \wedge \neg comeu(pedro, x)] \quad (2.10)$$

onde o objeto referenciado por maçã e o objeto referenciado por \emptyset são o mesmo, denotado por x .

No entanto, se o exemplo for ligeiramente modificando para

(2.11) Maria comprou [uma maçã], . Pedro não comprou \emptyset ,

esta identificação não pode ser feita. A maçã que Maria comprou e a que João deixou de comprar não são necessariamente a mesma. A representação lógica não pode identificá-las, sendo esta dada por uma expressão como

$$\exists x[maca(x) \wedge comprou(maria, x)] \wedge \neg \exists y[maca(y) \wedge comprou(pedro, y)] \quad (2.12)$$

Analisando superficialmente os dois exemplos, não é possível encontrar nenhuma diferença sintática entre eles. A mesma construção no sujeito, no predicado, verbos de transitividade iguais e até o mesmo número de palavras. Este exemplo constitui um problema para o interpretador de discursos, pois parece difícil saber quando as entidades devem ser igualadas ou não.

Na seção 4.1, mostro que (2.11) possui uma característica diferente dos demais. Embora este exemplo apresente uma referência indireta a um elemento extra-lingüístico, o que constitui um fenômeno fórico, não existe uma relação de correferência entre \emptyset e maçã. Mas precisa existir alguma relação entre eles, pois é o fato de Maria ter comprado uma maçã que permite uma interpretação para a segunda sentença. Isto pode ser observado omitindo-se a primeira sentença do exemplo.

(2.13) João não comprou.

Em (2.13), nada faz crer que João tenha deixado de comprar uma maçã. Este fato intrigante, a utilização de uma referência a um objeto para fazer referência indireta a um outro, ainda não referenciado no discurso, não é um tipo de fenômeno fórico comum.

Outro exemplo de discurso, que permite duas interpretações é apresentado abaixo.

(2.14) (a) Ana comprou um bolo, Maria vendeu \emptyset .

(b) Ana comprou [um bolo]_{*j*}, Maria vendeu \emptyset _{*j*}.³

O mesmo discurso permite duas interpretações diferentes para o elemento apagado. Na primeira delas, o complemento do verbo vendeu é o mesmo bolo que Ana comprou. Isto seria o mesmo que interpretar \emptyset como o apagamento do pronome oblíquo *o*.

$$\exists x[\text{bolo}(x) \wedge \text{comprou}(\text{ana}, x) \wedge \text{vendeu}(\text{maria}, x)]$$

Na segunda interpretação, um bolo é copiado para o lugar de \emptyset . Neste caso, a segunda sentença é interpretada como se fosse “complementar” à primeira: Maria vendeu um bolo em vez de comprar um bolo. Agora, o bolo comprado por Ana pode não ser o mesmo bolo que foi vendido por Maria.

$$\exists x[\text{bolo}(x) \wedge \text{comprou}(\text{ana}, x)] \wedge \exists y[\text{bolo}(y) \wedge \text{vendeu}(\text{maria}, y)]$$

No capítulo 4, mostro que para tratar exemplos como os dados acima, fui obrigado a criar um procedimento totalmente novo, baseado num sistema de cópia sintática.

³Neste texto, também utilizo as letras subscritas (*j*, neste caso) para denotar aquele tipo diferente de fenômeno fórico.

2.2.4 Papel dos prossintagmas

Um prossintagma verbal (ou prossintagma) é um elemento que precede o predicado da sentença. Normalmente, ele é colocado entre o sujeito e o verbo. Dentre estes, os que nos interessam nesta abordagem são os que parecem permitir o apagamento de partes do sintagma verbal. Como ilustração, considere o exemplo (2.3), repetido abaixo, no qual o apagamento do complemento verbal soa agramatical.

(2.15) *João comeu feijão_i. Maria comeu \emptyset _i.

Neste exemplo, a simples inserção de um prossintagma, neste caso um advérbio de negação, transforma-o em um discurso perfeitamente aceito.

(2.16) João comeu feijão_i. Maria não comeu \emptyset _i.

Em [25], Matos alerta para a importância de outros destes constituintes, como ‘também’, ‘também não’ e ‘sim’, no caso do PE. Mas sua teoria parece valer também no PB. Como pode ser visto a seguir.

(2.17) (a) João leu o livro_i. Maria também leu \emptyset _i.

(b) *João leu o livro_i. Maria leu \emptyset _i.

Estes constituintes permitem também o apagamento de segmentos complexos. (Os próximos exemplos ilustram este fato.

(2.18) João come [chocolate depois de lavar os dentes]_i. Ele tem um amigo que não come \emptyset _i.

(2.19) João [tinha lido estes livros para o filho]_i. Maria também tinha \emptyset _i.

(2.20) João gosta [de comer bolo no café]_i. Maria também gosta \emptyset _i.

A palavra ‘não’ pode aparecer na segunda sentença, como no exemplo (2.18) acima, ou na primeira, como no exemplo abaixo.

(2.21) João não come [chocolate depois de lavar os dentes]_i. Ele tem um amigo que come \emptyset _i.

No entanto, ela precisa estar presente. Caso contrário, o discurso não é aceito.

(2.22) *João come [chocolate depois de lavar os dentes]_i. Ele tem um amigo que come \emptyset _i.

Outro fato importante nestes exemplos é que a simples repetição do texto entre colchetes no lugar do \emptyset mantém a gramaticalidade e o significado das respectivas sentenças.

(2.18') João não come [chocolate depois de lavar os dentes]. Ele tem um amigo que come [chocolate depois de lavar os dentes].

(2.19') João tinha [lido estes livros para o filho]. Maria também tinha [lido estes livros para o filho].

(2.20') João gosta [de comer bolo no café]. Maria também gosta [de comer bolo no café].

Os apagamentos promovidos por estes constituintes podem se referir a objetos, como o livro no exemplo (2.17a), a segmentos complexos contendo adjuntos, como chocolate depois de lavar os dentes no exemplo (2.18), e a segmentos complexos contendo ações, como comer bolo no café no exemplo (2.20).

A exemplo do que ocorre em (2.11) e em (2.14b), os apagamentos vistos aqui não correferem com outros elementos lingüísticos do discurso. Também nestes casos, o tratamento deve ser feito de maneira diferenciada daquele adotado para resolver fenômenos fóricos comuns.

2.3 Considerações finais

Como pôde ser visto neste capítulo, o problema a ser solucionado relaciona-se exatamente com a característica que torna o objeto nulo um fenômeno diferente dos demais fenômenos fóricos. Encontrar elementos aos quais um constituinte ausente se refere não é a mesma coisa que encontrá-los para pronomes, por exemplo. Em primeiro lugar, como um pronome é explícito no discurso, pode-se ter mais certeza da sua função sintática. Logo, é possível agir com mais precisão na procura do objeto ao qual ele se refere. Pronomes trazem informações como gênero e número, por exemplo. Estas são pistas valiosas na resolução de fenômenos fóricos que os objetos nulos não trazem.

O presente trabalho busca soluções, ao menos parciais, para todos os problemas citados acima. No capítulo 4, apresento uma solução parcial para a representação do vazio, problema que provém da produção da palavra vazia em lugares indevidos. Apresento também uma solução satisfatória para as ambigüidades introduzidas pela dupla interpretação da transitividade dos verbos e pela identidade e não identidade entre objetos no discurso. Finalmente, objetos nulos acompanhados por prossintagmas verbais também recebem um tratamento.

De mais a mais, os vários resolvidores de fenômenos fóricos encontrados na literatura não apresentam procedimentos específicos para encontro do correferente de objetos nulos. Mais raros ainda são os que interpretam-no semanticamente, e nenhum deles faz isso no Português do Brasil.

Capítulo 3

Quadro teórico

As teorias de interpretação de discursos atuais estão fundamentadas sobre duas asserções básicas atribuídas ao matemático e filósofo alemão Gottlob Frege (1848–1925). Em primeiro lugar, Frege chegou à conclusão de que atribuir significado a uma sentença é o mesmo que descrever sob quais condições esta sentença seria verdadeira. Em segundo lugar, Frege também afirmou que, dada uma linguagem, a interpretação de construções complexas é determinada pela interpretação das partes individuais e pela maneira que estas partes são colocadas em conjunto. Esta última afirmação, conhecida como *Princípio da Composicionalidade*, aplicada ao contexto das linguagens naturais, propõe que o significado de uma sentença depende do significado das palavras envolvidas e da maneira com que estas palavras são colocadas em conjunto.

Tais asserções levaram a uma aproximação das teorias lingüísticas à lógica. Isto incentivou pesquisas no sentido de traduzir sentenças para linguagens formais de representação. Dentre as mais relevantes, estão os trabalhos de Richard Montague [27, 28] (veja também [20, 10]), da década de 70. Montague rejeitou a existência de uma distinção teórica importante entre a linguagem formal e a linguagem natural. Ele criou um homomorfismo (\mathcal{H}_1) de representações sintáticas de uma parte do Inglês a representações semânticas. E além deste, um outro homomorfismo (\mathcal{H}_2), agora das representações semânticas a interpretações em modelos teóricos. A composição $\mathcal{H} = \mathcal{H}_1 \circ \mathcal{H}_2$ é um homomorfismo direto da sintaxe aos modelos.

A exemplo de Montague, os procedimentos descritos neste capítulo constroem a interpretação de discursos de maneira composicional. As próximas seções fazem um apanhado das teorias utilizadas na implementação destes procedimentos.

3.1 Sintaxe

Nesta abordagem, vamos simplificar a noção de discurso definindo-o como uma simples seqüência de sentenças em linguagem natural. Sua interpretação consiste na modelagem

computacional de processos capazes de traduzí-los em algum formalismo lógico. Trata-se de uma tarefa semelhante a uma compilação, onde o código fonte seria a linguagem natural e o código objeto, o formalismo lógico.

A exemplo do que se faz em compiladores, a primeira providência a ser tomada é identificar o que deve ser aceito como uma instância bem formada do código fonte. Ou seja, o reconhecimento sintático. A rigor, o reconhecimento de uma instancia bem formada da linguagem natural deveria passar pelos níveis de reconhecimento fonético, morfológico e sintático. Todavia, o objetivo fixado aqui, a interpretação semântica de discursos focalizada na resolução dos fenômenos de apagamento do complemento verbal, não permitiu englobar todos os níveis de tratamento. Vale lembrar que o propósito desta pesquisa não visa a criação de uma ferramenta prática, e sim, a simples criação de um procedimento capaz de tratar o fenômeno. Desta forma, decidiu-se por excluir completamente os dois primeiros níveis e dar um tratamento ligeiramente superficial para o terceiro, concentrando os esforços nas camadas superiores.

Logo, o átomo da linguagem é a palavra ou, em alguns casos, um conjunto de palavras. Tomadas em conjunto, as palavra formam sintagmas. Os sintagmas formam sentenças que, juntas, formam um discurso completo.

A sintaxe descrita nesta seção foi inspirada na *Generalized Phrase Structure Grammar (GPSG)* descrita por Gazdar et al. [12]. Esta gramática foi escolhida, principalmente, por apresentar relativa facilidade de implementação no modelo computacional escolhido para construção do protótipo.

A GPSG é formada por regras cujos elementos possuem um conjunto de propriedades chamadas *estruturas de traços*. O léxico é definido separadamente e as regras são aplicadas de acordo com um sistema de regras de produção. Neste caso, porém, para que uma regra seja aplicada, as variáveis que aparecem dentro da estrutura de traços¹ devem ser *instanciadas* e seus valores *unificados* corretamente. A gramática designa, para o discurso reconhecido, uma árvore que mostra a divisão do discurso em constituintes sucessivamente menores rotulados pelo nome da categoria à qual pertencem. Como exemplo, considere o seguinte fragmento da gramática:

Definição 1 (Gramática 1)

$$(G1.1) \quad D \rightarrow S$$

$$(G1.2) \quad D \rightarrow S C D$$

$$(G1.3) \quad S \rightarrow \begin{array}{cc} SN & SV \\ \left[\begin{array}{c} num: \square \end{array} \right] & \left[\begin{array}{c} num: \square \end{array} \right] \end{array}$$

¹Simbolizadas com um número dentro de um pequeno quadrado.

$$(G1.4) \quad \begin{array}{l} SN \rightarrow N \\ \left[\begin{array}{l} num: \square \end{array} \right] \quad \left[\begin{array}{l} num: \square \end{array} \right] \end{array}$$

$$(G1.5) \quad \begin{array}{l} SV \rightarrow V \quad SN \\ \left[\begin{array}{l} num: \square \end{array} \right] \quad \left[\begin{array}{l} tipo: trans \\ num: \square \end{array} \right] \quad \left[\begin{array}{l} num: \square \end{array} \right] \end{array}$$

$$(G1.6) \quad \begin{array}{l} SV \rightarrow V \\ \left[\begin{array}{l} num: \square \end{array} \right] \quad \left[\begin{array}{l} tipo: intr \\ num: \square \end{array} \right] \end{array}$$

Entradas léxicas:

$$(G1.7) \quad \begin{array}{l} N \rightarrow \text{João} \\ \left[\begin{array}{l} num: sing \end{array} \right] \end{array}$$

$$(G1.8) \quad \begin{array}{l} N \rightarrow \text{Maria} \\ \left[\begin{array}{l} num: sing \end{array} \right] \end{array}$$

$$(G1.9) \quad \begin{array}{l} N \rightarrow \text{arroz} \\ \left[\begin{array}{l} num: sing \end{array} \right] \end{array}$$

$$(G1.10) \quad \begin{array}{l} V \rightarrow \text{ama} \\ \left[\begin{array}{l} tipo: trans \\ num: sing \end{array} \right] \end{array}$$

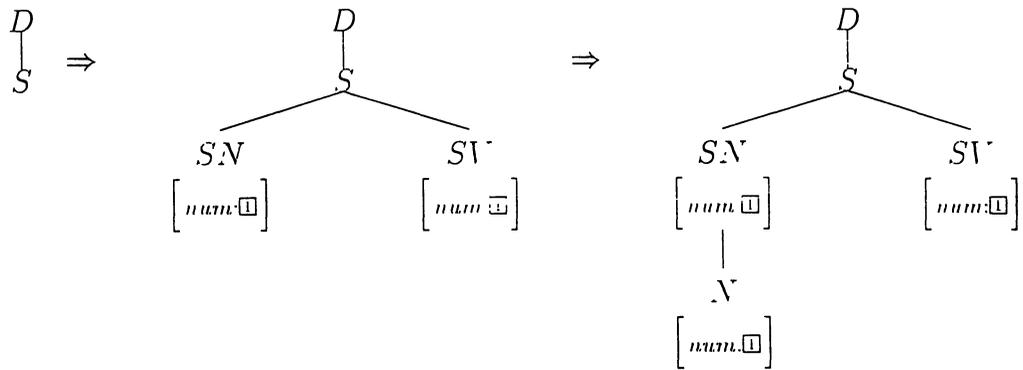
$$(G1.11) \quad \begin{array}{l} V \rightarrow \text{trabalhou} \\ \left[\begin{array}{l} tipo: intr \\ num: sing \end{array} \right] \end{array}$$

$$(G1.12) \quad C \rightarrow . \quad \blacksquare$$

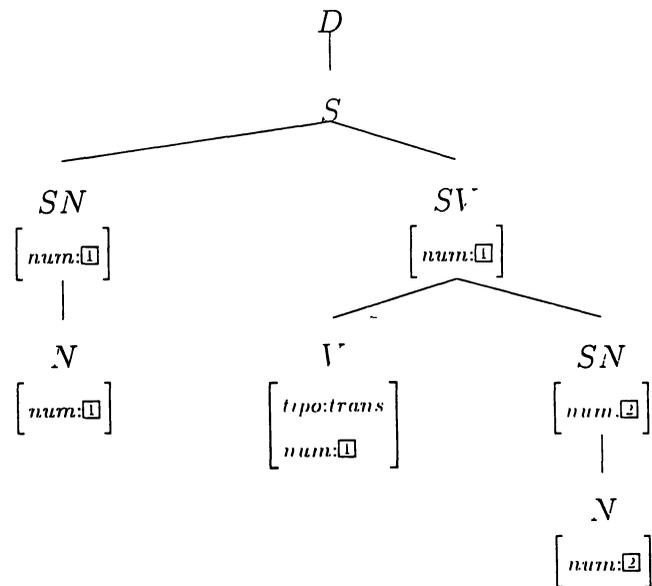
As duas primeiras regras desta gramática, (G1.1) e (G1.2), são o seu ponto inicial. Estas regras reconhecem discursos (D) recursivamente como sendo formados por uma ou mais sentenças (S) separadas por um conectivo (C). no caso da Gramática 1, o ponto. Uma sentença é formada por um sintagma nominal (SN) e um sintagma verbal (SV). O sintagma nominal é formado por um substantivo (N). O sintagma verbal constitui o predicado da sentença e é formado pelo verbo (V) e pelo complemento do verbo (se houver): um outro sintagma nominal. As entradas léxicas exemplificam como as palavras são reconhecidas pela gramática. Para exemplificar o funcionamento da gramática, considere

(3.1) João ama Maria.

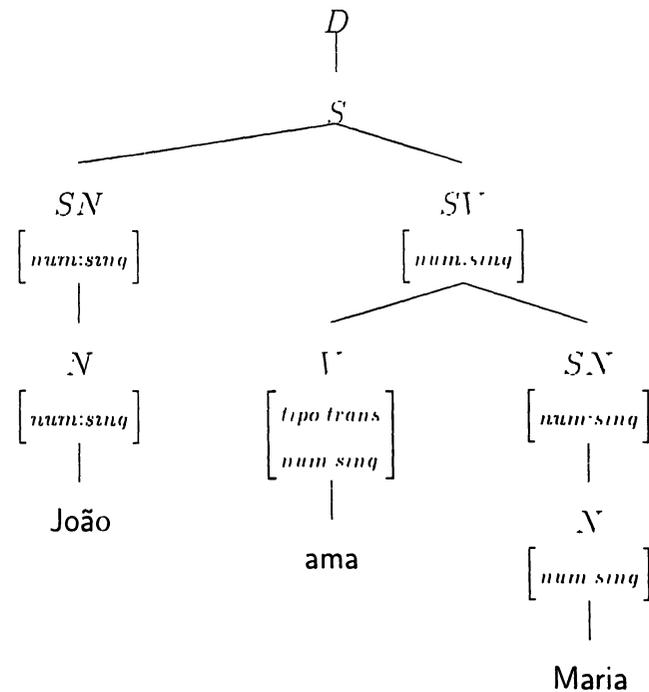
Para reconhecer (3.1), as regras de produção são sucessivamente aplicadas, como demonstrado abaixo.



Primeiro, a regra (G1.1) expande D e produz S , logo em seguida, a regra (G1.3) expande S produzindo os sintagmas nominal e verbal. A regra (G1.4) produz N expandindo SN .



A regra (G1.5) expande SV produzindo V' e SN e a regra (G1.4) é aplicada mais uma vez para expandir SN e produzir novamente um N . Note que neste ponto, as variáveis do primeiro SN e de SV já foram unificadas. Ambas são representadas pelo símbolo \square . Porém, o outro SN não possui sua variável unificada com aquelas, sendo esta representada por um símbolo diferente, \square .



A regra (G1.7) é aplicada e produz João, instanciando a variável num do N com o valor $sing$. Este valor é propagado através de SN e $SV̄$ até $V̄$. Como num também possui o valor $sing$ na regra (G1.10), ela pode ser aplicada para gerar *ama*. Finalmente, a regra (G1.8) é aplicada novamente para produzir *Maria*, o que também provoca a unificação das variáveis \square com o valor $sing$. Ao final de todo o processo, o discurso João ama Maria é reconhecido com sucesso.

3.2 Semântica

Segundo Eijck e Kamp, em [40], a relação semântica entre pronomes pessoais e seus antecedentes pode ser de dois tipos: um pronome funciona como uma referência a uma constante igual à de seu antecedente ou age como uma variável ligada pelo seu antecedente. Isto significa que pode-se representar um discurso como

(3.2) Pedro_i entrou. Ele_i sorriu.

de duas formas:

$$\text{entrou}(\text{pedro}) \wedge \text{sorriu}(\text{pedro}) \quad (3.3)$$

$$\text{entrou}(\text{pedro}) \wedge \text{sorriu}(x) \wedge x = \text{pedro} \quad (3.4)$$

Sabe-se também que é desejável que o procedimento obedeça ao princípio da composicionalidade mencionado anteriormente e seja, de certa forma, monotônico. Isto é, uma vez composta a representação de parte do discurso, esta não precise mais ser modificada. Isto é importante para garantir que a representação possa ser construída sistematicamente.

Estas características dependem, todavia, de um formalismo de representação que permita um sistema incremental de construção de representações de discursos.

Considere novamente o exemplo (3.2). Na tentativa de agir de forma sistemática, temos como representação dada para a primeira sentença, a fórmula

$$\text{entrou}(\text{pedro})$$

e para a segunda

$$\text{sorriu}(x)$$

considerando que o pronome Ele insere uma variável livre na representação. Seria, portanto, necessário igualar a variável x a *pedro* ou, alternativamente, substituí-la pela constante *pedro* para construir a representação completa do discurso.

Considere agora o exemplo

(3.5) Um homem, entrou. Ele, sorriu.

A primeira sentença deve ser representada por

$$\exists x[\text{homem}(x) \wedge \text{entrou}(x)]$$

e a segunda por

$$\text{sorriu}(y)$$

Seguindo o mesmo procedimento para construir a representação do discurso, temos

$$\exists x[\text{homem}(x) \wedge \text{entrou}(x)] \wedge \text{sorriu}(x) \quad (3.6)$$

ou

$$\exists x[\text{homem}(x) \wedge \text{entrou}(x)] \wedge \text{sorriu}(y) \wedge y = x \quad (3.7)$$

Nem no primeiro e nem no segundo caso, y representa o mesmo objeto que x . Este problema é chamado por Eijck e Kamp de *Problema da Ligação Anafórica do Contexto*. Observe que, a não ser que se utilize um procedimento bem mais complexo, não é possível a resolução do fenômeno fórico do exemplo (3.5). Pois, seria necessário aumentar o escopo do quantificador existencial adicionando o predicado $\text{sorriu}(y)$ e transformar a variável y em x para construir a representação correta, dada por

$$\exists x[\text{homem}(x) \wedge \text{entrou}(x) \wedge \text{sorriu}(x)] \quad (3.8)$$

Conclui-se, portanto, que a lógica de primeira ordem não favorece a construção da representação de discursos de forma composicional. Complementariamente, exemplos conhecidos como *donkey sentences*, apresentados por Geach em [13], demonstram que a

lógica de primeira ordem não captura o contexto das sentenças em um discurso.

(3.9) *If Pedro, owns a donkey, he, likes it.*

Se Pedro, tem uma "mula", ele, gosta dela.²

O exemplo (3.9) pode ser representado em lógica de primeira ordem como

$$\forall x((donkey(x) \wedge own(pedro, x)) \rightarrow like(pedro, x)) \quad (3.10)$$

No entanto, segundo [39], uma interpretação feita utilizando o procedimento criado por Montague ([27]) resultaria em

$$\exists x[(donkey(x) \wedge own(pedro, x)) \rightarrow like(pedro, x)] \quad (3.11)$$

ou em

$$\exists x[donkey(x) \wedge own(pedro, x)] \rightarrow like(pedro, x) \quad (3.12)$$

as quais são claramente interpretações errôneas. O problema reside na interpretação existencial atribuída ao *SN* indefinido, exatamente como em (3.6) e (3.7).

Portanto, utilizando a lógica de primeira ordem, o procedimento não é capaz de dar tratamento satisfatório a discursos com mais de uma sentença. Não é possível aproveitar o contexto das sentenças anteriormente analisadas na composição da representação da sentença seguinte. Esta é a motivação pela qual procurou-se por uma formalização alternativa que permitisse uma interpretação dinâmica. Ou seja, que reconsiderasse o conceito tradicional de quantificação de variáveis. No início dos anos 80, surge a *Discourse Representation Theory (DRT)* (Teoria de representação de discursos), desenvolvida por Kamp e Heim em [18, 16] (veja também [19]). Este formalismo provê uma concepção dinâmica do significado. Baseado na observação de que o ouvinte de um discurso é capaz de processá-lo de forma incremental, a DRT interpreta uma sentença utilizando o contexto estabelecido pelas sentenças anteriormente proferidas.

3.2.1 Teoria de representação de discurso

Como dito anteriormente, a interpretação de um discurso precisa descrever as situações em que o conteúdo deste é verdadeiro. Seguindo este raciocínio, e a exemplo da lógica de primeira ordem, a DRT implementa uma linguagem de representação lógica capaz de ser avaliada em um modelo teórico. Esta linguagem é construída com estruturas chamadas *Discourse Representation Structures (DRS)* (Estruturas de representação de Discurso).

²A tradução mais correta desta sentença seria: *Se pedro tem um burro, ele gosta dele*. Contudo, os dois pronomes masculinos impõem uma ambigüidade inexistente na versão em inglês. Desta forma, traduzi a sentença de uma forma um pouco diferente evitando assim o problema sem alterar o teor discussão que se segue.

Estas estruturas são formadas por uma tupla: $DRS = (U, Con)$, onde U é um conjunto que contém os *referentes de discurso*, as entidades do contexto atual e Con é um conjunto contendo *condições*, que denotam as propriedades e inter-relações entre as entidades.

Definição 2 (DRS)

Sejam a constante, u referente de discurso pertencente ao conjunto U , e R relação, então

termo: $t ::= u \mid a$

condição: $C ::= \top \mid R(t_1, \dots, t_k) \mid u = t \mid u \neq t \mid \neg D$

DRSs: $D ::= (\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\})$ ■

Por exemplo, a DRS que interpreta a sentença Ana trabalhou, é

$$(\{x_1\}, \{x_1 = ana, trabalhou(x_1)\}) \quad (3.13)$$

ou de forma mais amigável

x_1
$x_1 = ana$ $trabalhou(x_1)$

(3.14)

Tal representação é construída da seguinte forma: primeiramente, o SN do discurso determina a adição do referente x_1 em U e da condição $x_1 = ana$ em Con . Em seguida, o SV determina o surgimento da condição $trabalhou(x_1)$.

Adicionando, por exemplo, a sentença Ela se cansou ao discurso, as informações continuarão a ser incluídas na mesma DRS. A representação do discurso inteiro é

x_1, x_2
$x_1 = ana$ $trabalhou(x_1)$ $se_cansou(x_2)$ $x_2 = x_1$

(3.15)

Um novo referente é adicionado para o SN Ela e uma nova condição para o SV se cansou. A condição $x_2 = x_1$ é colocada na DRS porque x_2 representa um elemento fórico. Na DRT, cada referente introduzido por um elemento fórico deve ser identificado com um outro referente de discurso *acessível*. A acessibilidade é um conceito simples definido em termos de DRS aninhadas que será apresentado mais tarde.

As representações construídas com a DRT parecem diferir bastante da lógica de primeira ordem. Todavia, embora aquela seja uma representação mais “poderosa”, a idéia não é criar um formalismo totalmente distinto. Na verdade, a DRT se assemelha muito à lógica de primeira ordem, apenas a notação é diferente. Na DRT, a quantificação existencial

cial é implícita, ou seja, todo referente de discurso pode ser interpretado como quantificado existencialmente. A quantificação universal é representada utilizando-se a negação. Os operadores comumente utilizados naquela lógica (\wedge , \vee e \rightarrow) também estão presentes. A conjunção entre condições é representada pela simples inserção delas na DRS. Os outros dois operadores podem ser representados utilizando a conjunção e a negação. Como a representação poderia ficar muito carregada, foram definidas abreviações, as quais são apresentadas abaixo.

Definição 3 (Abreviações de DRSs)

Sejam D_1 e D_2 DRSs, então

$$\begin{aligned} (\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\}) \Rightarrow D_1 & ::= \neg(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m, \neg D_1\}) \\ D_1 \vee D_2 & ::= \neg(\{v_1, \dots, v_n\}, \{C_1, \dots, C_m, \neg D_1, \neg D_2\}) \quad \blacksquare \end{aligned}$$

Colocadas na forma de caixas, a implicação é abreviada como

$$\begin{array}{|c|} \hline v_1 \dots v_n \\ \hline C_1 \\ \vdots \\ C_k \\ \hline \end{array} \Rightarrow \begin{array}{|c|} \hline v_{n+1} \dots v_m \\ \hline C_{k+1} \\ \vdots \\ C_l \\ \hline \end{array} ::= \neg \begin{array}{|c|} \hline v_1 \dots v_n \\ \hline C_1 \\ \vdots \\ C_k \\ \hline \neg \begin{array}{|c|} \hline v_{n+1} \dots v_m \\ \hline C_{k+1} \\ \vdots \\ C_l \\ \hline \end{array} \\ \hline \end{array} \quad (3.16)$$

e, do mesmo modo, a disjunção como

$$\begin{array}{|c|} \hline v_1 \dots v_n \\ \hline C_1 \\ \vdots \\ C_k \\ \hline \end{array} \vee \begin{array}{|c|} \hline v_{n+1} \dots v_m \\ \hline C_{k+1} \\ \vdots \\ C_l \\ \hline \end{array} ::= \neg \begin{array}{|c|} \hline \neg \begin{array}{|c|} \hline v_1 \dots v_n \\ \hline C_1 \\ \vdots \\ C_k \\ \hline \end{array} \\ \hline \neg \begin{array}{|c|} \hline v_{n+1} \dots v_m \\ \hline C_{k+1} \\ \vdots \\ C_l \\ \hline \end{array} \\ \hline \end{array} \quad (3.17)$$

Como a DRS ambiciona capturar o significado de discursos, ela possui uma forma de descrever as condições em que determinado discurso é ou não verdadeiro. A exemplo da lógica de primeira ordem, a DRT possui uma definição de *satisfação*. Primeiramente, porém, deve-se utilizar uma forma de descrever a situação. Isso se faz através de modelos teóricos.

Definição 4 (Modelo)

Um **modelo** é o par ordenado $\mathcal{M} = (D, F)$, onde D é o conjunto de entidades chamado de **domínio** e F é a **função de interpretação** que especifica o valor semântico apropriado para cada entidade da seguinte forma:

1. Se c é uma constante, então $F(c) \in D$.
2. Se R é uma relação de n elementos, então $F(R)$ é um conjunto de tuplas formada por n elementos pertencentes a D . ■

Definição 5 (Atribuição)

Uma **atribuição** s de $\mathcal{M} = (D, F)$ é uma função parcial do conjunto de referentes \mathcal{U} a elementos de D . ■

Suponha que s e s' são atribuições em um mesmo modelo \mathcal{M} . Dizemos que s' *estende* s se e somente se sempre que $s(t)$ é definido, $s'(t)$ é definido também e $s(t) = s'(t)$. Sob a ótica da teoria de conjuntos, pode-se afirmar que $s \subseteq s'$. Intuitivamente, portanto, s' possui, no mínimo, tanta informação quanto s . Note também que toda atribuição estende a si mesma e que toda atribuição estende uma atribuição vazia.

Definição 6 (Satisfação de DRS)

Sejam \mathcal{M} um modelo e s uma atribuição. $\mathcal{M}, s \models (\{v_1, \dots, v_n\}, \{C_1, \dots, C_m\})$ sse existe uma atribuição s' tal que s' estende s e $s'(v_1), \dots, s'(v_n)$ são todos definidos e s' satisfaz todos os membros de $\{C_1, \dots, C_m\}$ em \mathcal{M} . ■

Intuitivamente, a definição dada acima significa que s relaciona os referentes de discurso com elementos do modelo de tal forma que todas as condições da DRS são satisfeitas. Contudo, a DRS principal pode conter condições complexas (sub-DRSs), as quais possuem novos referentes de discurso. Esta é a razão pela qual é necessário existir a possibilidade de estender a atribuição s .

Agora, é preciso definir o que significa satisfazer uma condição. A avaliação de um termo t determinada por \mathcal{M} e s , é definida pela função $I_{\mathcal{M},s}(t) ::= F(t)$ se t é uma constante e $I_{\mathcal{M},s}(t) ::= s(t)$ se t é um referente de discurso. Como $s(t)$ pode ser indefinido, $I_{\mathcal{M},s}(t)$ pode ser indefinido também.

Definição 7 (Satisfação de condição)

Sejam \mathcal{M} um modelo, s uma atribuição, v um referente, t um termo, D_1 e D_2 DRSs,

então

$$\mathcal{M}, s \models \top$$

$$\mathcal{M}, s \models R(t_1, \dots, t_n) \quad \text{sse} \quad I_{\mathcal{M},s}(t_1), \dots, I_{\mathcal{M},s}(t_n) \quad \text{são todos definidos e} \\ (I_{\mathcal{M},s}(t_1), \dots, I_{\mathcal{M},s}(t_n)) \in F(R)$$

$$\mathcal{M}, s \models v = t \quad \text{sse} \quad s(v) \text{ e } I_{\mathcal{M},s}(t) \text{ são definidos e } s(v) = I_{\mathcal{M},s}(t)$$

$$\mathcal{M}, s \models \neg D_1 \quad \text{sse} \quad \text{para todo } s' \text{ que estende } s \text{ em } D_1, s' \text{ não satisfaz } D_1 \text{ em } \mathcal{M}.$$

$$\mathcal{M}, s \models D_1 \vee D_2 \quad \text{sse} \quad s' \text{ satisfaz } D_1 \text{ em } \mathcal{M} \text{ ou } s'' \text{ satisfaz } D_2 \text{ em } \mathcal{M}$$

$$\mathcal{M}, s \models D_1 \Rightarrow D_2 \quad \text{sse} \quad \text{para todo } s' \text{ que estende } s \text{ e } s' \text{ satisfaz } D_1 \text{ em } \mathcal{M} \text{ existe} \\ \text{um } s'' \text{ que estende } s' \text{ e satisfaz } D_2 \text{ em } \mathcal{M}. \quad \blacksquare$$

Finalmente, podemos definir o que significa dizer que uma DRS é verdadeira.

Definição 8 (Condição verdade de DRS)

Uma DRS D é verdadeira em \mathcal{M} se existe uma atribuição que satisfaz D em \mathcal{M} . \blacksquare

Considere, como ilustração das definições dadas, um vocabulário definido por $V = \{(joao, 0), (maria, 0), (ana, 0), (ama, 2), (bonito, 1)\}$ e um modelo definido como

$$\mathcal{M} = (D, F)$$

$$D = \{d_1, d_2, d_3\}$$

$$F = \{(joao, d_1), \\ (maria, d_2), \\ (ana, d_3), \\ (ama, \{(d_1, d_2), (d_3, d_1)\}), \\ (bonito, \{d_1, d_2\})\}$$

Agora vamos tentar aplicar as definições para verificar se a DRS dada por

$$A = \begin{array}{|l} \cdot \quad x_1 \quad x_2 \\ \hline x_1 = joao \\ x_2 = maria \\ ama(x_1, x_2) \\ \hline \neg \\ \quad \begin{array}{|l} ama(x_2, x_1) \end{array} \end{array}$$

é verdadeira no modelo \mathcal{M} .

Esta DRS representa um discurso como João ama Maria. Ela não o ama. Vemos que tal discurso é verdadeiro no modelo. Logo, A também deve ser.

Para verificar que A é verdadeira em \mathcal{M} basta existir uma atribuição que satisfaça todas as condições de A . A atribuição s deve ser uma função parcial com domínio em U e contradomínio em D . Sendo assim, a defino como sendo $s = \{(x_1, d_1), (x_2, d_2)\}$. Uma vez definida a atribuição, verificamos se ela satisfaz as condições simples de A .

- $\mathcal{M}, s \models x_1 = joao$, pois $s(x_1) = d_1 = I_{\mathcal{M},s}(joao)$.
- $\mathcal{M}, s \models x_2 = maria$, pois $s(x_2) = d_2 = I_{\mathcal{M},s}(maria)$.
- $\mathcal{M}, s \models ama(x_1, x_2)$, pois $(I_{\mathcal{M},s}(x_1), I_{\mathcal{M},s}(x_2)) \in F(ama)$.

Para que s satisfaça a condição complexa de A , não pode existir uma atribuição s' que estenda s e que satisfaça a sub-DRS de A . Sabemos que $s \subseteq s'$. Logo $s'(x_1) = d_1$ e $s'(x_2) = d_2$, para qualquer s' que estenda s . Desta forma, temos que

$$\mathcal{M}, s' \not\models \frac{}{ama(x_2, x_1)} \quad \text{pois } (I_{\mathcal{M},s}(x_2), I_{\mathcal{M},s}(x_1)) = (d_2, d_1) \notin F(ama), \text{ para qualquer } s' \text{ que estende } s \text{ em } \mathcal{M}.$$

Portanto, $\mathcal{M}, s \models A$.

O exemplo acima apresenta um discurso contendo um pronome pessoal, *ela*. Para a resolução de fenômenos fóricos como este, a DRT utiliza com bastante frequência a acessibilidade, que é definida em termos da *subordinação*.

Definição 9 (Subordinação)

1. A DRS D_1 é **imediatamente subordinada** à DRS D_2 sse $D_1 = D_2$ ou $\neg D_1 \in Con_{D_2}$.
2. A DRS D_1 é **subordinada** à DRS D_2 sse
 - (a) D_1 é imediatamente subordinada a D_2 , ou
 - (b) existe uma DRS D_3 , a qual é subordinada a D_2 e D_1 é imediatamente subordinada a D_3 . ■

Definição 10 (Acessibilidade)

Sejam D_1 e D_2 DRSs. $u_1 \in U_{D_1}$ e $u_2 \in U_{D_2}$, então u_1 é acessível a u_2 sse D_2 é subordinada a D_1 . ■

A partir destas definições, é possível enxergar o conceito geométrico da acessibilidade na figura 3.1. Nesta figura estão representadas DRSs aninhadas. Condições presentes nos compartimento de onde partem as flechas têm acesso aos referentes pertencentes aos compartimentos onde chegam as flechas.

Para exemplificar o conceito, considere três exemplos.

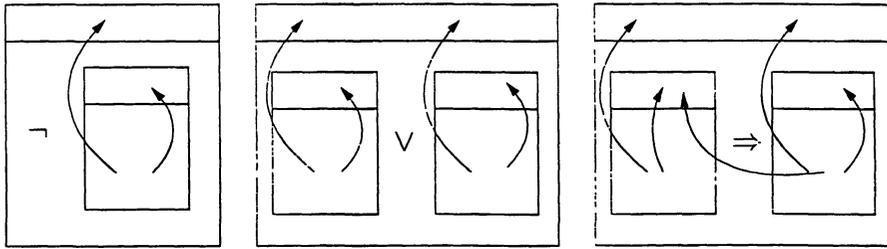


Figura 3.1: Ilustração geométrica do conceito de acessibilidade na DRT.

(3.18) Uma mulher trabalhou. Ela se cansou.

(3.19) João não comprou um bolo. Pedro o comeu.

(3.20) Se Pedro tem uma mula, ele gosta dela.

As representações das duas sentenças do exemplo (3.18) são dadas por

x_1
$mulher(x_1)$
$trabalhou(x)$

e por

x_2
$mulher(x_2)$
$se_cansou(x_2)$
$x_2 = ?$

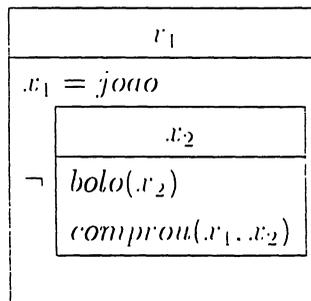
A representação de todo o discurso é dada pela a *fusão* das duas DRSs em apenas uma. Esta operação é feita realizando-se a união dos dois conjuntos de referentes e dos dois conjuntos de condições, formando uma única DRS.

$x_1 x_2$
$mulher(x_1)$
$trabalhou(x_1)$
$se_cansou(x_2)$
$x_2 = x_1$

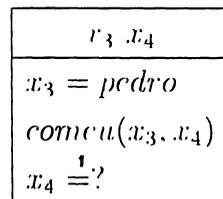
A DRT é capaz de identificar o referente x_2 com o referente x_1 porque x_1 está acessível àquele referente.

No exemplo (3.19), o pronome o não possui um correferente válido no discurso. Isto ocorre porque o bolo não existe. A DRT é capaz de capturar esta sutileza através das

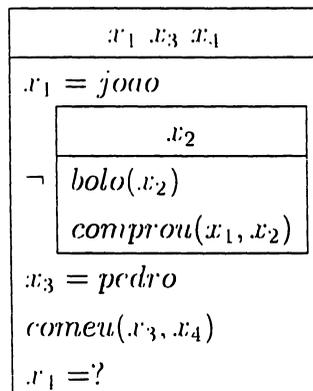
regras de acessibilidade. A representação da primeira sentença do exemplo é



e da segunda é

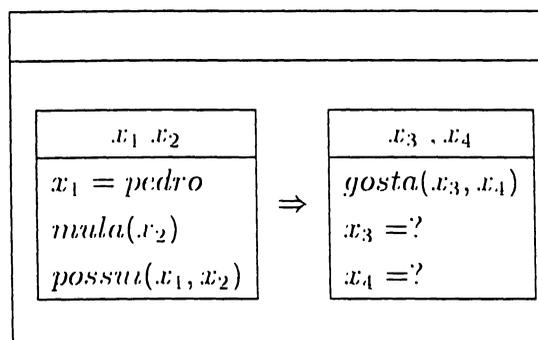


Novamente, a fusão das DRSs é feita para construir a representação do discurso todo.



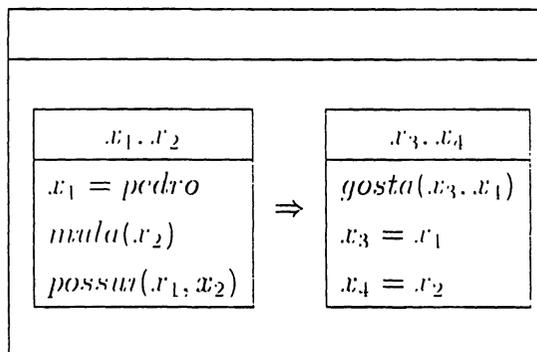
O referente x_4 não pode ser identificado com o referente x_2 , pois este não está acessível à DRS principal. O referente x_2 está acessível apenas à sua própria DRS.

O exemplo (3.20) é uma *donkey sentence*. A DRT consegue fazer a ligação dos coreferentes, impossível na lógica de primeira ordem. Abaixo, a primeira representação da sentença



A representação final é dada abaixo. Observe que os referentes x_1 e x_2 estão acessíveis aos referentes x_3 e x_4 , pois a DRS onde estes últimos aparecem está subordinada à DRS

principal pelo símbolo \Rightarrow :



Nos exemplos apresentados, a DRT se mostra capaz de simular o processamento incremental necessário à interpretação de discursos contendo referências fóricas. As regras de acessibilidade que tornam tal fato possível ainda restringem referências não aceitas pela linguagem, como apresentado no exemplo (3.19). Esta foi a grande motivação para a utilização desta teoria na presente pesquisa.

3.2.2 O cálculo lambda

O cálculo lambda foi desenvolvido na década de 30 pelo matemático Alonso Church ([5]) e constitui uma notação alternativa para a definição de funções. Neste formalismo, expressões são construídas com o *operador lambda* (λ) e podem denotar três tipos de coisas diferentes: *variáveis*, *aplicações* e *abstrações*.

Definição 11 (Expressão lambda)

Seja V uma variável e E, E_1, E_2 expressões lambda. então

$$\begin{array}{ll}
 E ::= V & | \text{ (variável)} \\
 & (E_1 E_2) \quad | \text{ (aplicação)} \\
 & \lambda V.E \quad \quad | \text{ (abstração)}
 \end{array}$$

Na aplicação, E_1 é chamado de **funtor** e E_2 de argumento. Na abstração, V é dita **ligada**. λV é chamado de **prefixo** e E é o **escopo** do operador lambda. ■

A abstração $\lambda V.E$ é uma função que toma um argumento a e retorna, como resultado, a função representada por E quando a variável ligada pelo operador lambda (V) vale a . Por exemplo, a expressão $\lambda x.x$ denota a identidade, pois se aplicada à uma expressão E qualquer, temos $((\lambda x.x)E) = E$.

Algumas abstrações lambda podem ser avaliadas através da aplicação de uma regra chamada *conversão beta*.

Definição 12 (Conversão beta)

Seja V uma variável e E_1, E_2 expressões lambda. então

$$(\lambda V.E_1)E_2 \xrightarrow{j} E_1[E_2/V]$$

Onde $E_1[E_2/V]$ é o resultado da substituição por E_2 de toda ocorrência de V em E_1 . ■

Com a conversão beta é possível utilizar o operador lambda como se fosse uma marca da falta de alguma informação. A abstração definida em (3.21) indica que a informação denotada pela variável x está ausente.

$$f = \lambda x.P(x) \quad (3.21)$$

Aplicada à constante a , f retorna, através da conversão beta, a expressão $P(a)$:

$$(\lambda x.P(x)) a = P(a)$$

Sendo P um predicado da lógica de primeira ordem e a uma constante, então se $a \in F(P)$, temos $f(a) = 1$, caso contrário $f(a) = 0$. Ou seja (3.21) é uma função dos objetos aos valores de verdade definida sobre o predicado P . Ou, mais precisamente, f é o conjunto dos objetos que possuem a propriedade P .

Algumas simplificações notacionais do cálculo lambda são freqüentemente utilizadas. Abaixo, elas são apresentadas.

Definição 13 (Convenções)

Sejam E_1, E_2, \dots, E_n expressões lambda, então

1. $E_1 E_2 \equiv (E_1 E_2)$
2. a aplicação é associativa à esquerda:
 $E_1 E_2 E_3 \dots E_n \equiv ((\dots((E_1 E_2)E_3)\dots)E_n)$
3. o alcance do operador de λV estende-se o mais à direita possível:
 $\lambda V.(E_1 E_2 \dots E_n) \equiv (\lambda V.(E_1 E_2 \dots E_n))$
4. a abstração é associativa à direita:
 $\lambda V_1 V_2 \dots V_n.E \equiv \lambda V_1.(\lambda V_2.(\dots(\lambda V_n.E)))$ ■

3.2.3 λ -DRS

Muskens, em [29], propôs um procedimento composicional para construção de DRSs a partir de discursos em linguagem natural. Ele recriou os homomorfismos outrora estabelecidos por Montague, agora utilizando como linguagem de representação a DRT. Para este propósito, Muskens utilizou, assim como Montague, o operador lambda e a conversão beta.

Muskens também propôs um operador capaz de unir duas DRSs em apenas uma, dando-lhe o nome de *operador fusão* (\oplus).

Definição 14 (Operador fusão)

Seja $D_1 = (\{v_1, \dots, v_n\}, \{C_1, \dots, C_l\})$ e $D_2 = (\{v_{n+1}, \dots, v_m\}, \{C_{l+1}, \dots, C_j\})$, então $D_1 \oplus D_2 ::= \{v_1, \dots, v_n, v_{n+1}, \dots, v_m\}, \{C_1, \dots, C_l, C_{l+1}, \dots, C_j\}$ ■

As informações semânticas dos constituintes lingüísticos podem agora ser compostas juntamente com o reconhecimento sintático do discurso, bastando acrescentar as representações semânticas nas palavras da linguagem. Abaixo, a gramática é redefinida para que isto possa ser feito.

Definição 15 (Gramática 2)

$$(G2.1) \quad D \rightarrow lfsSem:\boxed{1}$$

$$(G2.2) \quad D \rightarrow \begin{matrix} S & C & D \\ \left[\begin{matrix} sem:\boxed{2} & \boxed{1} & \boxed{3} \end{matrix} \right] & \left[\begin{matrix} sem:\boxed{1} \end{matrix} \right] & \left[\begin{matrix} sem:\boxed{2} \end{matrix} \right] & \left[\begin{matrix} sem:\boxed{3} \end{matrix} \right] \end{matrix}$$

$$(G2.3) \quad S \rightarrow \begin{matrix} SN & SV \\ \left[\begin{matrix} sem:\boxed{2} & \boxed{3} \end{matrix} \right] & \left[\begin{matrix} num:\boxed{1} \\ sem:\boxed{2} \end{matrix} \right] & \left[\begin{matrix} num:\boxed{1} \\ sem:\boxed{3} \end{matrix} \right] \end{matrix}$$

$$(G2.4) \quad SN \rightarrow \begin{matrix} N \\ \left[\begin{matrix} num:\boxed{1} \\ sem:\boxed{2} \end{matrix} \right] \end{matrix}$$

$$(G2.5) \quad SV \rightarrow \begin{matrix} V & SN \\ \left[\begin{matrix} num:\boxed{1} \\ sem:\boxed{2} & \boxed{4} \end{matrix} \right] & \left[\begin{matrix} tipo:trans \\ num:\boxed{1} \\ sem:\boxed{2} \end{matrix} \right] & \left[\begin{matrix} num:\boxed{2} \\ sem:\boxed{4} \end{matrix} \right] \end{matrix}$$

$$(G2.6) \quad SV \rightarrow \begin{matrix} V \\ \left[\begin{matrix} num:\boxed{1} \\ sem:\boxed{2} \end{matrix} \right] \end{matrix}$$

Entradas Léxicas:

$$(G2.7) \quad N \rightarrow \text{João}$$

$$\left[\begin{matrix} num:sing \\ sem:\lambda P. \left(\begin{matrix} x \\ x = \text{joão} \end{matrix} \right) \neg(Px) \end{matrix} \right]$$

$$(G2.8) \quad \begin{array}{c} N \\ \rightarrow \text{arroz} \end{array} \quad \left[\begin{array}{c} \text{num:sing} \\ \text{sem}:\lambda P \left(\begin{array}{c} x \\ \text{arroz}(x) \end{array} \right) \end{array} \right]$$

$$(G2.9) \quad \begin{array}{c} V' \\ \rightarrow \text{ama} \end{array} \quad \left[\begin{array}{c} \text{tipo:trans} \\ \text{num:sing} \\ \text{sem}:\lambda P\lambda x. \left(P\lambda y. \begin{array}{c} \\ \text{ama}(x,y) \end{array} \right) \end{array} \right]$$

$$(G2.10) \quad \begin{array}{c} V' \\ \rightarrow \text{trabalhou} \end{array} \quad \left[\begin{array}{c} \text{tipo:trans} \\ \text{num:sing} \\ \text{sem}:\lambda x. \begin{array}{c} \\ \text{trabalhou}(x) \end{array} \end{array} \right]$$

$$(G2.11) \quad C \quad \rightarrow \quad \left[\text{sem}:\lambda P\lambda Q.(P\oplus Q) \right]$$

■

Nas representações presentes na gramática acima, as variáveis em maiúsculas (P e Q) serão substituídas por fórmulas de segunda ordem enquanto que as minúsculas (x e y) por fórmulas de primeira ordem.

Para ilustrar a construção das representações com λ -DRS, considere novamente o exemplo João ama Maria. A árvore sintática da sentença é dada na figura 3.2, onde A é a representação semântica de João, B a representação de ama e C a representação de Maria, definidas pela Gramática 2 acima.

A representação semântica completa do discurso é dada pela composição das representações construída no nó D , dada por $A(BC)$. Primeiramente, é preciso realizar a conversão beta da expressão (BC) :

$$\begin{aligned} & \lambda P\lambda x. \left(P\lambda y. \begin{array}{c} \\ \text{ama}(x,y) \end{array} \right) \lambda P'. \left(\begin{array}{c} x' \\ x' = \text{maria} \end{array} \oplus (P'x') \right) \\ & \lambda x. \left(\lambda P'. \left(\begin{array}{c} x' \\ x' = \text{maria} \end{array} \oplus (P'x') \right) \lambda y. \begin{array}{c} \\ \text{ama}(x,y) \end{array} \right) \\ & \lambda x. \left(\begin{array}{c} x' \\ x' = \text{maria} \end{array} \oplus \left(\lambda y. \begin{array}{c} \\ \text{ama}(x,y) \end{array} x' \right) \right) \\ & \lambda x. \left(\begin{array}{c} x' \\ x' = \text{maria} \end{array} \oplus \begin{array}{c} \\ \text{ama}(x,x') \end{array} \right) \end{aligned}$$

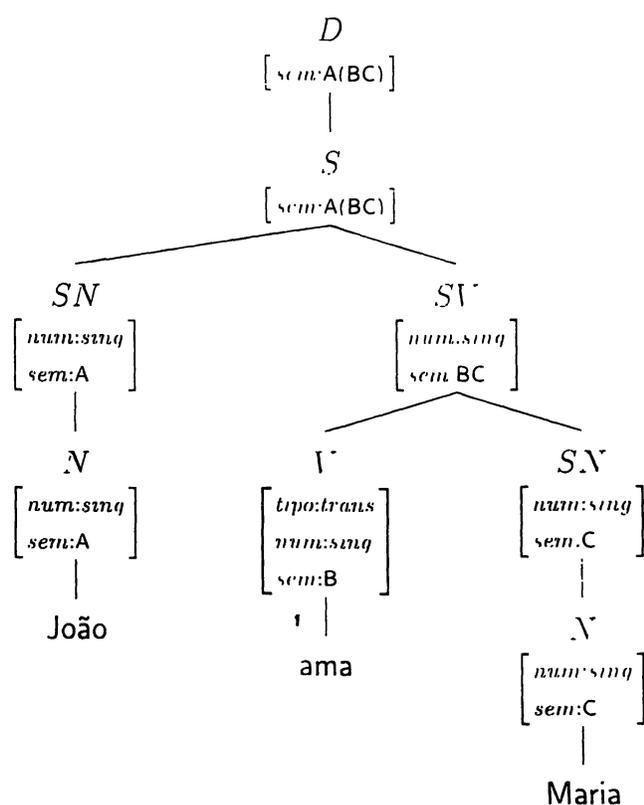


Figura 3.2: Árvore sintática: João ama Maria.

A operação de fusão é realizada de acordo com a definição 14 e a representação que corresponde ao SV da sentença é dada por

$$\lambda x. \begin{array}{|c|} \hline x' \\ \hline x' = maria \\ \hline ama(x, x') \\ \hline \end{array} \quad (3.22)$$

Esta, aplicada à representação semântica correspondente a João (A) resultará na representação final do discurso. A conversão beta é realizada abaixo.

$$\begin{array}{c}
\lambda P. \left(\begin{array}{|c|} \hline x'' \\ \hline x'' = \textit{jooao} \\ \hline \end{array} \oplus (Px'') \right) \lambda x. \begin{array}{|c|} \hline x' \\ \hline x' = \textit{maria} \\ \hline \textit{ama}(x, x') \\ \hline \end{array} \\
\\
\begin{array}{|c|} \hline x'' \\ \hline x'' = \textit{jooao} \\ \hline \end{array} \oplus \left(\lambda x. \begin{array}{|c|} \hline x' \\ \hline x' = \textit{maria} \\ \hline \textit{ama}(x, x') \\ \hline \end{array} x'' \right) \\
\\
\begin{array}{|c|} \hline x' \\ \hline x' = \textit{maria} \\ \hline \textit{ama}(x'', x') \\ \hline \end{array} \oplus \begin{array}{|c|} \hline x'' \\ \hline x'' = \textit{jooao} \\ \hline \end{array} \\
\\
\begin{array}{|c|} \hline x'' \ x' \\ \hline x'' = \textit{jooao} \\ \hline x' = \textit{maria} \\ \hline \textit{ama}(x'', x') \\ \hline \end{array}
\end{array}$$

Portanto, a λ -DRS torna possível construir a representação semântica de discursos em língua portuguesa com relevância contextual. No entanto, embora a DRT ofereça uma forma natural para tratamento de elementos fóricos da linguagem através da acessibilidade, este tratamento não é satisfatório. Em um discurso, é possível existir muitos candidatos a correferente. Neste caso, sua escolha deve ser criteriosa. Na seção seguinte são apresentadas algumas estratégias para fazer esta escolha.

3.3 Resolução de fenômenos fóricos

As regras que regem a resolução correta dos fenômenos fóricos são incrivelmente complexas e são objeto de estudo da lingüística e da computação há algum tempo. Várias heurísticas para sua resolução já foram desenvolvidas e são encontradas na literatura. Mitkov, em [26], fez um apanhado das mais utilizadas em modelos computacionais e classificou-as em dois grupos: *Restrições* e *Preferências*. O primeiro grupo é constituído de heurísticas que eliminam qualquer candidato a correferente que não cumpra determinadas regras. No segundo, os candidatos recebem um tipo de pontuação e aquele com melhor desempenho é escolhido. Como exemplo de restrições, podem ser listados gênero, número, consistência semântica e contra-índice. No grupo das preferências figuram paralelismo e saliência. As próximas seções descrevem as estratégias citadas e apresentam alguns exemplos.

3.3.1 Gênero e número

Gênero e número de elementos fóricos são informações bastante importantes no controle dos correferentes. Os próximos exemplos ilustram isto.

(3.23) (a) Maria, correu atrás de Pedro_j. Ela, não o_j, alcançou.

(b) Maria, e Paula_j, correram atrás de Pedro_k. Ele_k, não quis conversar com elas, _j.

(c) Maria_k, correu atrás de Pedro_i, e João_j. Eles, _j, não quiseram conversar com ela_k.

Observe que é possível trocar as posições dos correferentes dentro do discurso sem, contudo, transformá-lo em um discurso agramatical ou sem sentido. Não é possível, porém, utilizar um pronome masculino para referir-se a Maria, por exemplo. Da mesma forma, não se pode utilizar um pronome singular para referir-se, ao mesmo tempo, a João e a Pedro.

Utilizando somente estas duas restrições, é possível resolver as correferências corretamente nos exemplos acima. Mais do que isto, gênero e número são restrições que jamais podem ser violadas em uma referência fórica. Todavia, estas duas informações lexicais podem não ser suficientes, principalmente quando existem diversos candidatos a correferente no discurso.

3.3.2 Contra-índice

A restrição chamada de contra-índice baseia-se no fato de que vários verbos não podem ser reflexivos sem a presença de um pronome reflexivo. Assim como, se o seu objeto for um pronome reflexivo, este deve ser resolvido com o agente. Abaixo, ilustro com exemplos.

(3.24) (a) José, estava jogando futebol quando Pedro_j, o_j, viu no campo.

(b) José, estava jogando futebol quando Pedro_j, se_j, viu no espelho.

Como o objeto do verbo *ver* não pode ser o próprio agente se este não for um pronome reflexivo, a restrição de contra-índice não considera Pedro como um correferente possível em (3.24a). Em (3.24b), no entanto, o correferente precisa ser o agente. Neste caso Pedro deve ser escolhido.

3.3.3 Consistência semântica

A consistência semântica é uma restrição que se utiliza de algum conhecimento prévio acerca do significado dos constituintes lingüísticos. Este conhecimento pode ser atribuído a substantivos e verbos da linguagem através de propriedades que definem seus usos mais comuns. Para que dois elementos do discurso possam ser correferentes, tais propriedades devem ser consistentes entre si.

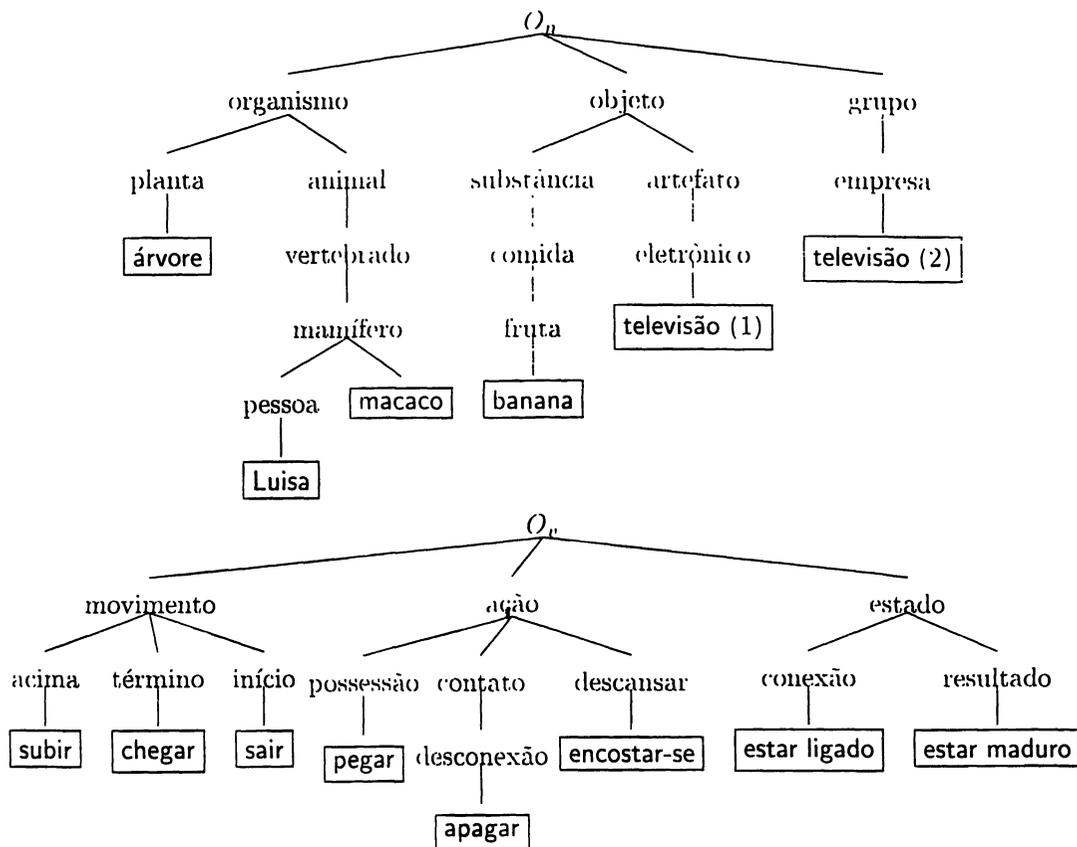


Figura 3.3: Ontologia de Saiz-Noeda e Palomar (O_n e O_v)

(3.25) A macaca subiu na árvore para pegar a banana. Ela, estava madura.

(3.26) A televisão, estava ligada quando Luisa chegou na cozinha. Ela a, apagou.

No exemplo (3.25), a propriedade de “estar maduro” é típica de uma fruta, e não de uma planta ou animal, razão pela qual *banana* é escolhido como correferente do pronome *ela*. No exemplo (3.26), por sua vez, “poder ser apagado” é uma propriedade normalmente atribuída a um equipamento eletrônico e não uma pessoa ou peça da casa. Por isso, a escolha de *televisão* como correferente de *a*.

Logo, uma representação do conhecimento de mundo, como a proposta de Saiz-Noeda e Palomar ([35]) para resolução de referência pronominais em espanhol, se faz necessária. Neste modelo, os substantivos e verbos da linguagem são classificados hierarquicamente em duas *ontologias* relacionadas através da compatibilidade entres seus elementos.

O modelo proposto é composto por quatro componentes, dois deles podem ser vistos na figura 3.3. O primeiro deles é a classificação ontológica dos substantivos (O_n). Esta ontologia é construída na forma de uma árvore. Cada nó folha representa um substantivo da linguagem. O conjunto de ancestrais de um dado substantivo n (A_n) representa suas características semânticas, organizadas da classe mais geral (nó raiz) para a mais específica (nós folha).

O segundo componente representa a ontologia dos verbos (O_v), seguindo o mesmo modelo proposto para O_n , classificando, portanto, todos os verbos da linguagem.

O terceiro componente (R_{sj}) define relações entre verbos e os substantivos que podem figurar como sujeito destes. O quarto componente (R_{ob}) define relações entre verbos e os substantivos que podem figurar como seus complementos. Estes dois componentes são formados por pares da forma (c_n, c_v) , onde $c_n \in O_n$ e $c_v \in O_v$.

No exemplo (3.25), a propriedade de "estar maduro" é decisiva para resolver a referência fórica do pronome *ela*. Observando novamente a figura 3.3, pode-se perceber que o substantivo *banana* é classificado como uma fruta, e esta classe possui uma relação de compatibilidade de complemento verbal com *estar maduro*. Ou seja, o par (fruta, estar-maduro) pertence ao conjunto R_{sj} . Semelhante fato não ocorre com os outros substantivos do discurso. Já no exemplo (3.26), a resolução da referência fórica utiliza o fato do par (eletrônico, desconexão) pertencer ao conjunto R_{ob} .

Através dos exemplos, é possível intuir quais seriam as definições de compatibilidade (denotadas pelo símbolo \in) entre um verbo e um substantivo nas posições de sujeito e de complemento.

Definição 16 (Compatibilidade entre verbo e sujeito)

Sejam v uma folha de O_v denotando um verbo, n uma folha de O_n denotando um substantivo, A_v o conjunto de ancestrais de v e A_n o conjunto de ancestrais de n , então $n \in_{sj} v$ sse $\exists(c_n, c_v) \in A_n \times A_v$ tal que $(c_n, c_v) \in R_{sj}$. ■

Definição 17 (Compatibilidade entre verbo e complemento verbal)

Sejam v uma folha de O_v denotando um verbo, n uma folha de O_n denotando um substantivo, A_v o conjunto de ancestrais de v e A_n o conjunto de ancestrais de n , então $n \in_{ob} v$ sse $\exists(c_n, c_v) \in A_n \times A_v$ tal que $(c_n, c_v) \in R_{ob}$. ■

Este processo de verificação de compatibilidade pode ser incorporado ao processo de resolução de fenômenos fóricos, onde as informações provenientes da compatibilidade entre verbos e substantivos pode ser aproveitada.

3.3.4 Paralelismo

Esta é uma heurística bastante simples que determina que o correferente de um pronome deve ser preferencialmente escolhido dentre os substantivos que desempenham a mesma função sintática nas sentenças anteriores. Como ilustração, considere o exemplo abaixo.

- (3.27) (a) Vicente entregou o disco a Sérgio, Carlos entregou a ele, uma carta.
 (b) Vicente, entregou o disco a Sérgio. Ele, entregou a Carlos uma carta.

Os correferentes do exemplo (3.27a) denotam quem sofre a ação do verbo *dar*, enquanto que em (3.27b), denotam quem a executa.

3.3.5 Saliência

Os critérios sintáticos e semânticos constituem um conjunto de heurísticas bastante eficazes na seleção de possíveis candidatos a correferente de elementos fóricos. Porém, em alguns casos, eles não são capazes de selecionar apenas um candidato para cada ocorrência fórica, o que provoca uma ambigüidade. Com o objetivo de evitar casos como este, utilizam-se alguns algoritmos baseados em teorias de saliência, capazes de descobrir quais são os elementos mais salientes do discurso atual, dando a estes a preferência na resolução da ocorrência fórica.

De acordo com as teorias de saliência, um discurso deve apresentar coerência na seqüência de enunciados que o forma. Um discurso deve exibir coerência global (entre os seus diversos segmentos) e coerência local (entre as declarações de um mesmo segmento). Logo, estas teorias propõe modelos que descrevem sistemas de regras que governam as relações entre o foco de atenção do discurso e as declarações que o compõem. Ilustro os conceitos descritos através de dois exemplos adaptados de Grosz et al. ([15]):

- (3.28) (i) João foi a sua loja de música favorita para comprar um piano.
 (ii) Ele havia freqüentado a loja por vários anos.
 (iii) Estava excitado porque iria finalmente poder comprar um piano.
 (iv) Mas quando chegou, a loja estava fechada.

- (3.29) (i) João foi a sua loja de música favorita para comprar um piano.
 (ii) Esta era a loja que João freqüentou por vários anos.
 (iii) Ele estava excitado porque iria finalmente poder comprar um piano.
 (iv) Ela estava fechada quando João chegou.

Os dois segmentos de discurso expressam exatamente a mesma informação utilizando enunciados diferentes. No entanto, o discurso (3.28) é intuitivamente mais coerente que o discurso (3.29). Isso acontece porque no primeiro caso trata-se apenas de um indivíduo central, João, enquanto que no segundo, o foco principal oscila entre João e a loja de música. Isso mostra que diferentes formas sintáticas implicam em diferenças na inferência dos correferentes fóricos para o leitor (ou ouvinte). Tanto a teoria de *centering*, proposta originalmente por Grosz et al. em 1983 e publicada bem mais tarde, em 1995 [15], quanto a teoria de *focusing* proposta por Sidner em [38] e complementada por ela mesma em [36, 37], fornecem bases para tratamento destas diferenças.

As próximas seções descrevem estas teorias apresentando algoritmos que as implementam na prática. Além disso, é feita também uma pequena comparação entre duas implementações criadas.

Focusing

A teoria de *focusing* tenta prever quais são as entidades mais prováveis a serem referidas nas sentenças subsequentes. A sintaxe e a semântica dos elementos é utilizada apenas para confirmar ou rejeitar a escolha. O elemento de maior saliência do discurso é chamado nesta abordagem de *foco*. O foco é o objeto sobre o qual a atenção de quem fala está centralizada no momento. Este elemento é um dos importantes detalhes que tornam uma série de sentenças peças coerentes de um texto ou discurso. Um discursante pode proferir várias sentenças sobre um único foco sem reintroduzi-lo em sua forma sintática completa. Para tanto, ele usa elementos fóricos. Existem, porém, duas espécies de foco, *foco ator* (f_a) e *foco de discurso* (f_d). Para distingui-las, considere o exemplo (3.30).

(3.30) Pedro mostrou um relógio. O relógio é bonito. Ele gosta dele.

As três sentenças do exemplo trazem informações sobre o elemento que recebe o foco do discurso, o relógio. Ele é introduzido por um artigo indefinido na primeira sentença e referenciado por O relógio e dele nas sentenças seguintes. Note que existe um outro substantivo introduzido no texto, Pedro, que é referenciado na terceira sentença pelo pronome Ele. Este é o foco ator. Observe que ambos desempenham papéis importantes, porém, um deles, o foco de discurso, sempre sofre a ação e o outro, o foco ator, é quem sempre desempenha a ação. Por esta razão, elementos fóricos localizados na posição de agente da sentença tendem a se referir aos focos atores, enquanto que elementos localizados em outras posições tendem a se referir ao foco de discurso.

O foco pode se deslocar de um objeto para outro no decorrer do discurso. No exemplo

(3.31) Pedro cozinhou arroz. Ele cozinhou feijão e o comeu.

o primeiro foco de discurso é arroz, mas a sentença seguinte introduz outro foco, feijão. Observe que o pronome o refere-se a feijão, que é o foco atual. Os focos comportam-se como em uma estrutura de pilha. O segundo foco é empilhado sobre o primeiro e no momento da resolução da ocorrência fórica, o objeto localizado no topo da pilha possui a preferência. Se alguma restrição impedir sua utilização, o elemento no sub-topo passa a ser preferido e assim por diante, até o fim da pilha.

Por vezes, o foco ator pode ocupar o lugar do foco de discurso. Todavia, isto ocorre somente em casos em que alguma restrição impede uma referência a este.

(3.32) Pedro, beijou Maria e João não falou mais com ele.

Observe que o pronome *ele* está em posição de objeto, mas se refere ao foco ator, Pedro. Como o pronome é masculino, ele não poderia se referir ao foco de discurso, Maria. Logo, se nenhum elemento da pilha dos focos de discurso pode resolver a ocorrência, a outra pilha, a de focos atores, passa a possuir a preferência.

Seguindo estas observações, é possível confeccionar um algoritmo que implemente a teoria de *focusmg*, a exemplo do que foi feito em [3]. Este algoritmo deve ser executado para cada sentença do discurso realizando três passos distintos:

Definição 18 (Algoritmo de *focusmg*)

1. Se for a primeira sentença do discurso, escolha os primeiros focos ator e de discurso;
2. Utilize os focos para resolver as ocorrências fóricas;
3. Atualize os focos ao final de cada sentença. ■

O primeiro passo inicializa as pilhas de focos do discurso. O primeiro foco ator é facilmente determinado, basta escolher o agente da primeira sentença. Determinar o foco de discurso, em geral, é mais problemático. É necessário descobrir qual é o assunto principal do discurso. A grande dificuldade, no entanto, reside no fato de que muitos fenômenos da linguagem podem indicar este elemento. Simplificadamente, porém, o algoritmo identifica o foco baseado somente nos verbos. Para esta abordagem, o foco de discurso será o objeto do verbo principal da sentença.

O segundo passo resolve as ocorrências fóricas utilizando os focos. O algoritmo executa esta tarefa seguindo as duas regras abaixo:

Definição 19 (Algoritmo de escolha dos referentes)

1. Se o elemento fórico desempenha função de objeto, escolha como correferente um dos elementos do foco de discurso.
2. Se o elemento fórico desempenha função de agente, escolha um dos elementos do foco ator como correferente. ■

Estas regras serão aplicadas se as outras restrições, como gênero, número, contra-índice e consistência semântica não rejeitarem a escolha. Caso haja uma rejeição, o elemento fórico será resolvido com o primeiro elemento da pilha que respeita todas as restrições. Novamente, o algoritmo baseia-se nos verbos para determinar a função desempenhada pelos elementos fóricos.

Após a análise de cada sentença, o foco precisa ser atualizado. A atualização do foco ator é feita da seguinte forma:

Definição 20 (Algoritmo de atualização dos focos atores)

1. Se a sentença possui um novo agente, adicione-o à pilha de focos atores;
2. Caso contrário, nada precisa ser feito. ■

A atualização do foco de discurso, por sua vez, segue as seguintes regras:

Definição 21 (Algoritmo de atualização dos focos de discurso)

1. Se já existe um foco de discurso e uma correferência fórica ligando-o, então o foco fica como estava;
2. Caso contrário, se a sentença possui um objeto, adicione-o na pilha de focos de discurso;
3. Caso contrário, nada precisa ser feito. ■

Centering

Na teoria de *centering*, um segmento de discurso consiste de uma seqüência de enunciados U_1, \dots, U_n . Os enunciados possuem a propriedade de *realizar* entidades do contexto do discurso. Por exemplo, o enunciado João foi a sua loja de música favorita comprar um piano, possui como entidades realizadas: joao, loja-de-musica e piano.

Os *centros* representam as entidades do mundo referidas pela sentença atual. Estes objetos servem de ligação entre um enunciado e outro no segmento de discurso que os contém.

Para cada enunciado U_n é associado um conjunto ordenado de *centros prospectivos* (*Forward-looking centers*) ($C_f(U_n)$), consistindo das entidades do discurso que são realizadas por este enunciado. Estes são ordenados conforme sua preferência a serem referenciados no próximo enunciado. Por ser o centro preferido, o primeiro elemento desta lista é chamado de *próximo centro preferencial* (*Preferred center*), ($C_p(U_n)$). O enunciado *está centrado*, isto é, discorre sobre uma única entidade chamada de *centro retrospectivo* (*Backward-looking center*), ($C_b(U_n)$), que é realizado pelo enunciado imediatamente anterior, U_{n-1} . Abaixo, um exemplo que serve como ilustração dos termos introduzidos

- (3.33) (i) João_{a₁} possui um BMW_{a₂}.
- (ii) Ele_{r₁} dirige rápido.
- (iii) Pedro_{a₃} corre com ele_{r₂} no feriado_{a₄}.
- (iv) Ele_{r₃} freqüentemente o_{r₁} vence.

Como pode ser observado no exemplo, as entidades do mundo são reconhecidas através dos substantivos que as descrevem: João. BMW. Pedro e feriado. Os índices que aparecem nos enunciados (a_1 , a_2 , a_3 e a_4) nomeiam as construções do segmento de discurso que referenciam as entidades. Da mesma forma, nomeia-se os elementos fóricos encontrados (r_1 , r_2 , r_3 e r_4). Com a classificação completada, pode-se construir a *âncora*, que é o par $\langle C_b, C_f \rangle$ de cada enunciado.

- (i) $\langle (?), [(joao, a_1), (bwm, a_2)] \rangle$
- (ii) $\langle [(joao, a_1)], [(joao, r_1)] \rangle$
- (iii) $\langle [(joao, r_1)], [(pedro, a_3), (joao, r_2), (feriado, a_4)] \rangle$
- (iv) $\langle [(pedro, a_3)], [(pedro, r_3), (joao, r_4)] \rangle$

O C_b apenas indica quem é a atual entidade central do discurso. O conjunto C_f por sua vez, indica quais entidades foram realizadas por quais elementos. O par (pedro, r_3) da âncora (iv), por exemplo, indica que a entidade pedro foi realizada pelo elemento r_3 . A primeira âncora apresenta um (?) como C_b . Isto acontece porque a teoria não define como escolher o C_b do primeiro enunciado do segmento de discurso.

Outro conceito importante da teoria são as transições entre os enunciados. Elas descrevem como estes são ligados em um segmento de discurso coerente. Grosz et al. ([15]) propuseram três tipos de transições. No entanto, o algoritmo construído posteriormente por Brennan et al. ([4]) propõe quatro tipos. Segundo eles, se o falante possui um certo número de proposições a expressar, ele faz isto coerentemente expressando todas as proposições sobre a primeira entidade (*continuing*) depois introduz uma entidade relacionada (*retaining*) e em seguida muda a atenção para outra entidade (*shifting*). O falante pode mudar para uma entidade que será o centro preferido do próximo enunciado (*smooth-shifting*) ou para um centro que logo perderá a preferência (*rough-shifting*).

Abaixo, um exemplo com os tipos de transições descritos. Os enunciados (iii-a), (iii-b), (iii-c) e (iii-d) são possíveis continuações do discurso formado pelos enunciados (i) e (ii).

- (3.34) (i) Pedro $_{a_1}$ é um cantor.
 $\langle (?), [(pedro, a_1)] \rangle$
- (ii) Ele $_{r_1}$ visitou Maria $_{a_2}$ ontem.
 $\langle [(pedro, a_1)], [(pedro, r_1), (maria, a_2)] \rangle$
- (iii-a) Ele $_{r_2}$ conversou muito com ela $_{r_3}$.
Continuing: $\langle [(pedro, r_1)], [(pedro, r_2), (maria, r_3)] \rangle$
- (iii-b) Ela $_{r_2}$ recebeu a visita dele $_{r_3}$ entusiasmada.
Retaining: $\langle [(pedro, r_1)], [(maria, r_2), (pedro, r_3)] \rangle$

(iii-c) Ela_{r₂} não gostou.

Smooth-shifting: $\langle (\text{maria}, a_2), [(\text{maria}, r_2)] \rangle$

(iii-d) Joana_{a₃} a_{r₂} viu na semana passada.

Rough-shifting: $\langle (\text{maria}, a_2), [(\text{joana}, a_3), (\text{maria}, r_2)] \rangle$

Comparação entre os algoritmos

A comparação que faço nesta seção se dá entre dois algoritmos implementados em outra oportunidade. Um deles implementa o algoritmo de *focusing* descrito na seção 3.3.5 e é completamente descrito em [22]. O outro implementa o algoritmo de *centering* descrito na seção 3.3.5 e é completamente descrito em [23].

A comparação feita com os dois algoritmos pretendia apontar diferenças na resolução dos pronomes das sentenças na medida que a escolha do elemento mais saliente do discurso se faz de maneira diferenciada. Para que esta comparação fosse justa, utilizei exatamente a mesma gramática e o mesmo conhecimento de mundo nas duas implementações.

A primeira diferença notada entre os algoritmos é que embora a teoria de *centering* pareça ser mais “elegante”, a característica combinatorial dos cálculos empregados descrevem um algoritmo muito menos eficiente do que *focusing*, ao menos do ponto de vista de recursos computacionais. *Centering* também demonstrou possuir alguns pequenos problemas contornáveis, pelo menos aparentemente. Dentre eles, a inicialização, a definição dos limites de cada segmento e referências “distantes”.

Focusing não apresenta o problema da escolha dos elementos salientes do primeiro enunciado do discurso (inicialização). Deste modo, discursos contendo elementos fóricos já na primeira sentença podem ser resolvidos por este algoritmo, excluindo-se aqueles que referem-se a elementos ainda não introduzidos no discurso. *Focusing* também possui a vantagem de não observar apenas a sentença imediatamente anterior para detectar os elementos centrais do discurso, o que o habilita a resolver ocorrências “distantes”.

O resultado final dos testes, porém, não apresenta grande diferenças entre as duas teorias. Posso dizer que, até onde pude verificar, a implementação utilizando *centering* não perde em muito para a feita com o algoritmo de *focusing*. Vale lembrar que o conjunto de testes permitidos pelo dicionário e gramáticas implementados não é muito grande. Além disso, ele não abrange uma parcela considerável de casos da linguagem. Por estas razões, para solução do objeto nulo, decidi utilizar a implementação feita com *focusing*.

3.3.6 α -DRT

Para a resolução dos fenômenos fóricos, estendi a linguagem adicionando um *operador alfa* (α), exatamente como Blackburn e Bos em [3]. O novo termo foi chamado de α -DRS. O operador α liga variáveis que representam referências fóricas contidas no

discurso. Seguindo na computação do discurso, o algoritmo proposto deve substituir as variáveis ligadas por α por referentes de discurso acessíveis e que respeitem algumas restrições impostas pela variável. As restrições são provenientes de informações captadas do próprio elemento fórico que a variável denota, ou de outros elementos do discurso. Para exemplificar, considere a sentença Ele trabalhou. Esta é representada como

$$\alpha a_{ele} \begin{array}{|c|} \hline a \\ \hline \text{trabalhou}(a) \\ \hline \end{array} \quad (3.35)$$

O operador α liga a variável a , que deverá ser substituída por um candidato a correferente. O referente escolhido deve ser compatível com as restrições da variável a . Em (3.35), as restrições são impostas pelo elemento fórico que a denota, além das informações provenientes das condições que possuem a na DRS. Um candidato a correferente só pode ser escolhido para a se denotar um objeto masculino singular. Além disso, o pronome Ele não é reflexivo e é sujeito do verbo trabalhar.

O operador α , portanto, replica a estratégia usada com o cálculo lambda, marcando a “informação ausente” a ser preenchida mais tarde. As representações semânticas dos elementos fóricos da linguagem fazem uso deste operador, como pode ser visto nas adições feitas à Gramática 2 listadas abaixo.

Definição 22 (Adendos à Gramática 2)

$$(G2.12) \quad S \rightarrow Z \quad SV^*$$

$$\left[\begin{array}{l} \text{sem:} \boxed{1} \ \boxed{2} \end{array} \right] \quad \left[\begin{array}{l} \text{sem:} \boxed{1} \end{array} \right] \quad \left[\begin{array}{l} \text{num:} \boxed{1} \\ \text{sem:} \boxed{2} \end{array} \right]$$

$$(G2.13) \quad N \rightarrow Pro$$

$$\left[\begin{array}{l} \text{num:} \boxed{1} \\ \text{sem:} \boxed{2} \end{array} \right] \quad \left[\begin{array}{l} \text{num:} \boxed{1} \\ \text{sem:} \boxed{2} \end{array} \right]$$

$$(G2.14) \quad Pro \rightarrow \text{ele}$$

$$\left[\begin{array}{l} \text{num:} \text{sing} \\ \text{sem:} \lambda P \alpha a_{ele}(Pa) \end{array} \right]$$

$$(G2.15) \quad Z \rightarrow \epsilon$$

$$\left[\text{sem:} \lambda P \alpha a_{zero}(Pa) \right]$$

■

As novas regras da gramática acima exemplificam como discursos contendo referências fóricas são reconhecidos. As regras (G2.13) e (G2.14) reconhecem pronomes pessoais como substitutos de substantivos. Da mesma forma, pode-se fazer o reconhecimento de

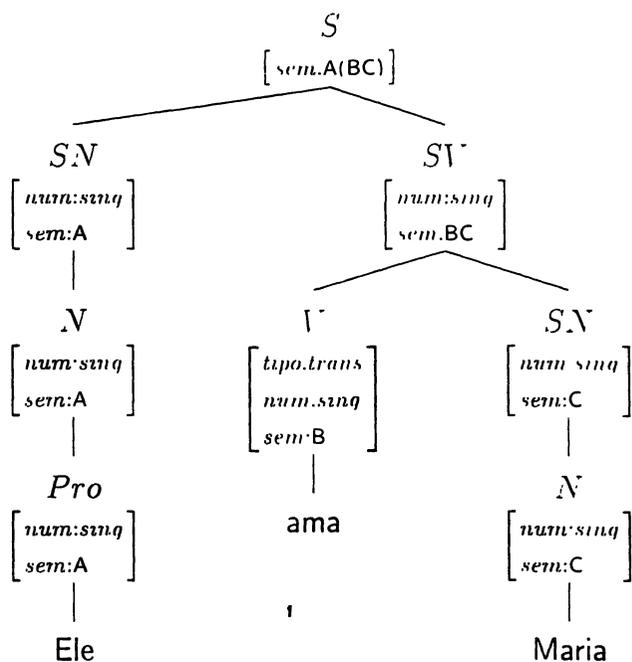


Figura 3.4: Árvore sintática: Ele ama maria.

pronomes oblíquos. A regra (G2.12), por sua vez, permite o aparecimento da chamada *zero-anáfora* em posição de sujeito. Vê-se através desta e da regra (G2.15), que a ausência de sujeito é tratada como se fosse um pronome.

Como ilustração do funcionamento desta gramática, considere uma sentença que contenha um elemento fórico, como no exemplo abaixo.

(3.36) Ele ama Maria

A árvore sintática dada pelo reconhecedor sintático é apresentada na figura 3.4. O passo seguinte realiza a conversão beta das representações, retornando a α -DRS da sentença. Primeiramente apresento novamente o resultado de (BC), dado em (3.22).

$$\lambda x. \begin{array}{|c|} \hline x' \\ \hline x' = maria \\ \hline ama(x, x') \\ \hline \end{array}$$

Esta DRS, aplicada à representação semântica correspondente a Ele (A) resultará na representação final da sentença. A conversão beta é realizada abaixo.

$$\lambda P \alpha_{ele}. (Pa) \left(\lambda x. \begin{array}{|l} x' \\ \hline x' = maria \\ \hline ama(x, x') \end{array} \right)$$

$$\alpha_{ele}. \left(\lambda x. \begin{array}{|l} x' \\ \hline x' = maria \\ \hline ama(x, x') \end{array} a \right)$$

$$\alpha_{ele}. \begin{array}{|l} x' \\ \hline x' = maria \\ \hline ama(a, x') \end{array}$$

3.3.7 Juntando tudo

Com a α -DRT, as teorias apresentadas para resolução de fenômenos fóricos podem ser colocadas em conjunto. Como ilustração do processo de representação de discursos com elementos fóricos, considere o exemplo abaixo.

(3.37) João comprou arroz. Cozinhou-o e almoçou.

Em 3.37 é preciso resolver duas ocorrências fóricas que aparecem na segunda sentença. A primeira delas, chamada sujeito nulo, ocorre porque existe um agente “subentendido” que executa a ação de cozinhar e almoçar. O correferente escolhido deverá ser o substantivo João. A segunda ocorrência é introduzida pelo pronome oblíquo *o* que se refere a arroz. O verbo cozinhar é reconhecido da mesma forma que o verbo amar e o verbo almoçar da mesma forma que o verbo trabalhar. A representação da primeira sentença é

$x_1 \ x_2$
$x_1 = joao$
$arroz(x_2)$
$comprou(x_1, x_2)$

enquanto que a representação da segunda sentença é dada por

$a \ b$
$cozinhou(a, b)$
$almocou(a)$

Combinando as duas representações chega-se à expressão

$$\alpha a_z \alpha b_o \quad \begin{array}{|l} x_1 \ x_2 \ a \ b \\ \hline x_1 = \text{jooao} \\ \text{arroz}(x_2) \\ \text{comprou}(x_1, x_2) \\ \text{cozinhou}(a, b) \\ \text{almocou}(a) \end{array} \quad (3.38)$$

Para encontrar o correferente de uma variável v ligada pelo operador α , utiliza-se o seguinte método:

1. Escolha um referente u acessível à DRS atual para candidato a correferente de v .
2. Verifique se as restrições de gênero, número e contra-índice são satisfeitas. Se alguma restrição não for satisfeita, volte ao passo 1.
3. Verifique se para cada condição $R_1(t_1, \dots, t_n)$, onde $t_i = u$ para algum $1 \leq i \leq n$ e para cada condição $R_2(t_1, \dots, t_m)$ tal que $t_j = v$ para algum $1 \leq j \leq m$, se $R_2 \in R_1$. Se em algum caso não houver compatibilidade, então volte ao passo 1.
4. Se não houver mais candidatos a escolher, v não possui correferente válido.

Para a verificação de compatibilidade entre duas condições R_1 e R_2 , propus o seguinte algoritmo:

Definição 23 (Compatibilidade entre condições)

1. Se R_1 representa um verbo e t_i está em posição de agente ($i = 1$) e R_2 representa um substantivo (neste caso temos $R_2(t_j)$), então $R_2 \in R_1 \leftrightarrow R_2 \in_{s_j} R_1$.
2. Se R_2 representa um verbo e t_i está em posição de objeto ($i > 1$) e R_1 representa um substantivo, então $R_2 \in R_1 \leftrightarrow R_2 \in_{ob} R_1$. ■

Em (3.38), x_2 não pode ser correferente de a porque existe uma incompatibilidade entre as condições $\text{arroz}(x_2)$ e $\text{cozinhou}(a, b)$. No modelo de conhecimento de mundo utilizado, arroz não pode cozinhar. No entanto, x_1 não apresenta incompatibilidade com a restrição de a , sendo este escolhido como seu correferente. Analogamente, x_2 é escolhido como correferente de b . O último passo liga as variáveis com os correferentes e a representação final do discurso fica

$x_1 \ x_2 \ a \ b$
$x_1 = joao$
$arroz(x_2)$
$comprou(x_1, x_2)$
$cozinhou(a, b)$
$almocou(a)$
$a = x_1$
$b = x_2$

Para levar em conta a saliência dos elementos do discurso, o algoritmo descrito aqui deve fazer uso do algoritmo de *focusing*, descrito na seção 3.3.5. Por esta razão, os focos de discurso e ator são distingüidos dos demais referentes nas DRSs. Criou-se então uma estrutura chamada de *foco-DRS* ([3]). Esta possui dois novos compartimentos para abrigar os focos. A sentença Pedro cozinhou arroz, por exemplo, é representada pela foco-DRS da seguinte maneira:

$x_1 \ x_2$
x_1
x_2
$x_1 = pedro$
$arroz(x_2)$
$cozinhou(x_1, x_2)$

O primeiro compartimento desta DRS é o compartimento dos referentes de discurso, o segundo passa a ser agora o compartimento de focos atores enquanto que o terceiro o de focos de discurso. No exemplo dado, x_1 é o foco ator, Pedro, e x_2 o foco de discurso, arroz. Obviamente, no decorrer do discurso, podem haver mais focos e a atenção pode se deslocar de um para outro.

Suponha agora que o discurso anterior possua mais uma sentença, por exemplo, Ele o comeu. A expressão que representa esta sentença é dada por

$\alpha a_e \alpha b_o$	
	$comeu(a, b)$

Combinando esta representação à da primeira sentença, temos

x_1, x_2
x_1
x_2
$\alpha a_e, \alpha b_o$ $x_1 = pedro$ $arroz(x_2)$ $cozinhou(x_1, x_2)$ $comeu(a, b)$

Examinando as condições da α -DRS, temos que a é um agente e b é um objeto. Dessa forma, as regras são capazes de resolver os elementos fóricos corretamente. O correferente escolhido para a é x_1 , que é o foco ator, enquanto que correferente de b é o foco de discurso x_2 . A representação final é

x_1, x_2
x_1
x_2
$x_1 = pedro$ $arroz(x_2)$ $cozinhou(x_1, x_2)$ $comeu(a, b)$ $a = x_1$ $b = x_2$

Lembre-se, porém, que as restrições das variáveis ligadas pelo operador α devem ser levadas em conta. Considere o exemplo Maria ama Pedro. Ele gosta dela. A expressão que o representa é

x_1, x_2
x_1
x_2
$\alpha a_e, \alpha b_{ela}$ $x_1 = maria$ $x_2 = pedro$ $ama(x_1, x_2)$ $gosta(a, b)$

Este exemplo ilustra um caso em que as restrições da variável a não permitem que esta seja correferenciada ao foco ator. Da mesma forma b não pode ser correferenciado ao foco de discurso. Em ambos os casos, a preferência passa a ser dada para a outra pilha. Agora, os papéis dos focos se invertem e a representação do discurso fica

$v_1 \ x_2$
$v_2 \ v_1$
$v_1 \ x_2$
$x_1 = \text{maria}$
$x_2 = \text{pedro}$
$\text{ama}(x_1, x_2)$
$\text{gosta}(a, b)$

Finalmente, formalizo o algoritmo completo de resolução de fenômenos fóricos:

Definição 24 (Algoritmo de resolução de fenômenos fóricos – \mathcal{F})

Seja uma variável v , ligada por α , então

1. Escolha o candidato a correferente de v :
 - (a) Se v está na posição de agente da sentença, ou se não há mais focos de discurso a serem escolhidos, escolha o próximo referente u da pilha de focos atores.
 - (b) Se v está na posição de objeto do verbo da sentença ou se não há mais focos atores a serem escolhidos, escolha o próximo referente u da pilha de focos de discurso.
 - (c) Se não há mais focos a serem escolhidos, v não possui correferentes válidos no discurso.
2. Verifique se as restrições de gênero, número e contra-índice são satisfeitas. Se alguma delas não for satisfeita, volte ao passo 1.
3. Verifique se para cada condição $R_1(t_1, \dots, t_n)$, onde $t_i = u$ para algum $1 \leq i \leq n$ e para cada condição $R_2(t_1, \dots, t_m)$ tal que $t_j = v$ para algum $1 \leq j \leq m$, se $R_2 \in R_1$. Se em algum caso não houver compatibilidade, então volte ao passo 1. Caso contrário, u é escolhido como correferente de v .
4. Atualize os focos atores e de discurso com as entidades da sentença atual. ■

Como última observação ao algoritmo, lembre que qualquer referente acessível pode ser candidato a correferente. Ou seja, se v está em uma DRS aninhada, então as pilhas de focos da DRS principal também podem ser utilizadas.

3.4 Considerações finais

Segundo Mitkov ([26]), todas as tentativas computacionais para tratamento de fenômenos fóricos utilizam de uma forma ou de outra os fatores acima. Existem também

algumas tentativas probabilísticas e emprego de técnicas de inteligência artificial que, todavia, não superam as anteriores em termos de resultados.

A busca por uma ferramenta que trate estes fenômenos engloba a utilização dos fatores certos na ordem de precedência correta. Além disso, a maioria esmagadora das pesquisas no campo dos fenômenos fóricos e da interpretação semântica de discursos foi feita para o Inglês. Apenas alguns estudos apontam avanços em outras línguas como Francês, Italiano, Espanhol ou Japonês. Pouquíssimos trabalhos foram desenvolvidos tendo como objeto de estudos a língua portuguesa. Alguns exemplos são: o trabalho de Paraboni [31, 32], que desenvolveu um protótipo que resolve referências pronominais possessivas em Português utilizando algumas heurísticas e conhecimento de mundo. No entanto, não chega a interpretar as sentenças e traduzi-las para algum formalismo lógico. Na mesma linha, o trabalho de Petry [33] avaliou o desempenho do algoritmo de *centering* na resolução de fenômenos fóricos pronominais em pequenos trechos de histórias infantis. Abraços e Lopes [1], por sua vez, utilizam o algoritmo de *focusmg* juntamente com a DRT para resolver pronomes e elipses. Finalmente, eu desenvolvi um protótipo para interpretação semântica de uma pequena parte do Português e que resolve alguns casos de referências pronominais, definidas e elipses [22].

Embora muitas tentativas de resolver o correferente para estes fenômenos tenham obtido bons resultados, até o momento nenhuma delas solucionou 100% dos casos. Também há provas conclusivas de que todos os fatores importantes inerentes a isto já tenham sido descobertos.

Capítulo 4

Nova proposta para o objeto nulo

A solução dada ao fenômeno objeto nulo, apresentada neste capítulo, resolve um problema mais amplo, onde a zero-anáfora se combina com outro fenômeno, a elipse de *SV*. Esta solução, contudo, surgiu da tentativa de explicar alguns casos de apagamento acompanhados por constituintes especiais, chamados por Ilari ([17]) de prossintagmas verbais. Como apresentado no capítulo 2, tanto Matos ([25]) como Ilari afirmam que estes constituintes parecem permitir o apagamento de partes de uma sentença. Os exemplos são repetidos aqui para melhor visualização.

- (4.1) (a) João não come [chocolate depois de lavar os dentes]_i. Ele tem um amigo que come \emptyset_i .
(b) João não come [chocolate depois de lavar os dentes]. Ele tem um amigo que come [chocolate depois de lavar os dentes].
- (4.2) (a) João tinha [lido estes livros para o filho]_j. Maria também tinha \emptyset_j .
(b) João tinha [lido estes livros para o filho]. Maria também tinha [lido estes livros para o filho].
- (4.3) (a) João gosta [de comer bolo no café]_k. Maria também gosta \emptyset_k .
(b) João gosta [de comer bolo no café]. Maria também gosta [de comer bolo no café].

Novamente, observe que a simples substituição de \emptyset pelo texto entre colchetes mantém as sentenças sintaticamente corretas e com o mesmo sentido. Lembre que tais ocorrências não são fenômenos fóricos, pois não há um objeto que correferre com \emptyset .

Outro problema a ser resolvido é a dupla interpretação de um mesmo discurso. Este fato, diretamente relacionado com a identidade dos objetos representados, pode ser ilustrado pelo exemplo (2.14), repetido abaixo.

- (4.4) (a) Ana comprou um bolo_i. Maria vendeu \emptyset_i .

(b) Ana comprou [um bolo]_j. Maria vendeu \emptyset _j.

Em (4.4a), o bolo vendido por Maria deve ser o mesmo que Ana comprou, enquanto que em (4.4b) ele pode não ser o mesmo.

Alguns verbos permitem interpretações intransitivas, como o exemplo (2.8), também repetido abaixo.

(4.5) João foi ao restaurante com Maria. Ela não comeu.

Finalmente, algumas ocorrências de apagamento são agramaticais ou, mesmo sendo aceitas pela gramática, não constituem referência a elemento algum. O discurso (2.3) repetido aqui, exemplifica um caso destes.

(4.6) *João comeu feijão_j. Maria comeu \emptyset _j.

Portanto, se a pretensão é construir um procedimento capaz de resolver objetos nulos, duas providências se fazem necessárias. Em primeiro lugar, é necessário elaborar as regras que habilitam ou não sua ocorrência. Aplicadas, estas regras devem impor interpretações corretas em cada um dos casos tratados pelo procedimento. Segundo, é preciso separar os tipos de fenômeno. Sabe-se que existem no mínimo dois. Um deles pode ser tratado através de um mecanismo de cópia, enquanto que o outro deve ser resolvido como um fenômeno fórico.

As próximas seções apresentam as soluções propostas para cada um destes problemas. Ao final do capítulo, uma seção apresenta algumas das limitações do procedimento adotado.

4.1 Condições para apagamento do objeto

Vários estudos realizados por lingüistas (Cyrino [9, 24], Omena [30], Matos [25] e Duarte [11]) apontam características que podem ser utilizadas como regras de habilitação do objeto nulo. Estes estudos apontam algumas restrições a alguns elementos da sentença que contêm o fenômeno e aos correferentes do objeto apagado. Por exemplo, Cyrino em [9], apoiando-se em uma pesquisa quantitativa, propõe que os objetos só podem ser apagados se possuírem correferentes inanimados. Além disso, como a própria Cyrino comenta, sua teoria é confirmada pelo trabalho anterior de Duarte ([11]). Dois de seus exemplos são transcritos abaixo para ilustrar.

(4.7) João descascou a banana, mas Pedro não comeu \emptyset _j. ([9] p.210)

(4.8) *João trouxe Maria_j mas Pedro não beijou \emptyset _j. ([9] p.211)

O substantivo *banana* do exemplo (4.7) denota uma entidade inanimada, o que permite seu apagamento na sentença subsequente. No exemplo (4.8) o único elemento que poderia ter sido apagado é animado. Por esta razão, este discurso não possui uma interpretação possível.

Outra regra proposta por Cívino diz respeito à diferenciação entre zero-anáforas e elipses de *SV*. No primeiro caso, o objeto referenciado pelo elemento elidido também é referenciado por outro constituinte no discurso. Perfazendo, portanto, um caso de correferência fórica. O segundo caso constitui uma referência a um objeto totalmente novo, mas que possui uma certa relação com algum elemento anteriormente referenciado. O contraste entre estas duas formas de apagamento foi rerepresentada há pouco. Os exemplos (4.4a) e (4.4b) ilustram bem o fato. Suas respectivas representações em lógica de primeira ordem e em DRT são dadas abaixo.

$$\exists x[\text{bolo}(x) \wedge \text{comprou}(\text{ana}, x) \wedge \text{vendeu}(\text{maria}, x)]$$

x_1 x_2 x_3
$x_1 = \text{ana}$
$\text{bolo}(x_2)$
$\text{comprou}(x_1, x_2)$
$x_3 = \text{maria}$
$\text{vendeu}(x_3, x_2)$

$$\exists x[\text{bolo}(x) \wedge \text{comprou}(\text{ana}, x)] \wedge \exists y[\text{bolo}(y) \wedge \text{vendeu}(\text{maria}, y)]$$

x_1 x_2 x_3 x_4
$x_1 = \text{ana}$
$\text{bolo}(x_2)$
$\text{comprou}(x_1, x_2)$
$x_3 = \text{maria}$
$\text{bolo}(x_4)$
$\text{vendeu}(x_3, x_4)$

Como visto, os fenômenos são de origem diferente. O primeiro deles comporta-se como uma zero-anáfora. Logo, pode ser tratado através da produção da palavra vazia (ϵ), como discutido nas seções 2.2.1 e 3.3. Leva-se em conta, evidentemente, as restrições semânticas que o fenômeno impõe. O outro fenômeno caracteriza-se pelo apagamento de parte do *SV* de uma sentença, o qual é reconstruído à partir de uma sentença anterior. Este, portanto, necessita de um tratamento novo, o qual é descrito adiante.

A questão que deve ser respondida agora é: como identificar uma ocorrência validada de elipse de *SV*? Para tentar descobrir, vamos construir todas as possibilidades de ausência de parte do *SV* de uma sentença e tentar encontrar um padrão.

- (4.9) João odeia comer arroz.
- | | | | | |
|-----|-----------|-------|----------|---------|
| (a) | Maria | odeia | cozinhar | ∅. |
| (b) | *Maria | odeia | ∅ | feijão. |
| (c) | Maria, | ∅ | comer | feijão. |
| (d) | Maria | adora | ∅ | ∅. |
| (e) | Maria, | ∅ | cozinhar | ∅. |
| (f) | Maria, | ∅ | ∅ | feijão. |
| (g) | Maria não | ∅ | ∅ | ∅. |

Observe que é possível o apagamento de quase qualquer parte do SV . Todavia, o apagamento só pôde ser feito porque existe uma certa "complementariedade" entre o SV da primeira sentença e o SV da segunda. Em (4.9a), por exemplo, quando se diz João odeia comer arroz. Maria odeia cozinhar, o significado soa como se em vez de "comer arroz", o que Maria odeia é "cozinhar arroz".

Observe que parece não ser possível apagar o SV da segunda sentença mantendo as mesmas informações do SV da primeira. Abaixo, apresento exemplos.

- (4.10) João odeia comer arroz.
- | | | | | |
|-----|---------|-------|-------|--------|
| (a) | *Maria | odeia | comer | ∅. |
| (b) | *Maria | odeia | ∅ | arroz. |
| (c) | *Maria, | ∅ | comer | arroz. |
| (d) | *Maria | odeia | ∅ | ∅. |
| (e) | *Maria, | ∅ | comer | ∅. |
| (f) | *Maria, | ∅ | ∅ | arroz. |
| (g) | *Maria | ∅ | ∅ | ∅. |

Nenhum dos SV 's incompletos de (4.10) é gramatical assumindo a interpretação na qual as elipses estão presentes. Ou seja, alguns exemplos podem tomar outra interpretação. Por exemplo, se considerarmos odiar ou comer como verbos intransitivos em (4.10a), (4.10d) e (4.10e), ou se não considerarmos o apagamento do verbo comer em (4.10b), temos discursos gramaticais. Aliás, talvez isto esteja relacionado à agramaticalidade do exemplo (4.9b). Como é possível interpretar este exemplo sem a existência da elipse, esta talvez seja a interpretação "mais natural", impedindo outra interpretação.

Logo, para que exista elipse de SV , é necessária a presença da complementariedade. A definição de complementariedade, contudo, é pragmática. É possível, empiricamente, mostrá-la através dos exemplos dados. Veja que em (4.9a) e (4.9e), o verbo comer e adquire sentido "complementar" a cozinhar. Não seria possível manter a gramaticalidade do discurso substituindo o verbo comer por vender, por exemplo. Em (4.9c) e (4.9f), a complementariedade fica por conta do substantivo feijão no lugar de arroz. Em (4.9d), o verbo adorar aparece exatamente como o contrário de odiar. O exemplo (4.9g), apresenta o modificador não invertendo o sentido do verbo odiar.

Existe, porém, uma forma de quebrar esta regra. Ou seja, existe uma forma de fazer o SV da segunda sentença do discurso possuir exatamente o mesmo significado do SV da primeira. E isto é feito com a inserção do prossintagma também, como nos exemplos abaixo.

- (4.11) João odeia comer arroz. (a) Maria também \emptyset \emptyset \emptyset .
 (b) Maria também odeia \emptyset \emptyset .
 (c) Maria também odeia comer \emptyset .

Portanto, é possível definir a regra que permite o apagamento de *SV*'s ou parte deles. Além disso, implícito a esta regra está o algoritmo de reconstrução dos *SV*'s "incompletos":

Definição 25 (Regra de apagamento de *SV*)

Todo *SV* parcialmente apagado pode copiar a parte faltante de um *SV* anterior no discurso ao qual apresente um sentido 'complementar', a não ser que o *SV* incompleto seja precedido pelo prossintagma 'também'. Neste caso, o sentido deve ser exatamente o mesmo. ■

A implementação computacional deste procedimento precisaria levar em consideração aspectos pragmáticos para conseguir discernir quando dois *SV*'s possuem sentidos complementares. Por ser muito custoso e estar fora do escopo da pesquisa realizada, esta característica não foi implementada completamente. O que foi feito foi a verificação de verbos e substantivos quanto a sua classificação ontológica. Adiante, apresento esta proposta com mais detalhes.

4.2 Um algoritmo para resolução do fenômeno

Nesta seção, apresento um algoritmo para tratamento parcial do objeto nulo e da elipse de *SV* no Português do Brasil. Este algoritmo aproveita as idéias já introduzidas pela literatura, principalmente na área da lingüística, apresentadas nas seções 3.3 e 4.1.

Quando é detectada a ausência de algum elemento na posição de objeto, a primeira coisa a ser feita pelo algoritmo é a distinção entre os fenômenos possíveis. Lembre-se que o tratamento da zero-anáfora é diferente do tratamento a ser dado para a elipse de *SV*. Contudo, na seção 2.2.3, apresentei um exemplo que pode ser fonte de problemas neste processo. Este exemplo é repetido aqui:

- (4.12) (a) Ana comprou um bolo_i. Maria vendeu \emptyset _i.
 (b) Ana comprou [um bolo]_j. Maria vendeu \emptyset _j.

As duas interpretações possíveis são agora dadas em DRT.

$$\begin{array}{cc}
 \text{(a)} & \text{(b)} \\
 \begin{array}{|l}
 \hline
 x_1 \ x_2 \ x_3 \\
 \hline
 x_1 = \textit{ana} \\
 \textit{bolo}(x_2) \\
 \textit{comprou}(x_1, x_2) \\
 x_3 = \textit{maria} \\
 \textit{vendeu}(x_3, x_2) \\
 \hline
 \end{array} & \begin{array}{|l}
 \hline
 x_1 \ x_2 \ x_3 \ x_4 \\
 \hline
 x_1 = \textit{ana} \\
 \textit{bolo}(x_2) \\
 \textit{comprou}(x_1, x_2) \\
 x_3 = \textit{maria} \\
 \textit{bolo}(x_4) \\
 \textit{vendeu}(x_3, x_4) \\
 \hline
 \end{array} \\
 & \text{(4.13)}
 \end{array}$$

Conclui-se, então, que os dois fenômenos não são mutuamente exclusivos. Ou seja, um discurso pode apresentar os dois ao mesmo tempo. Obviamente, se isso ocorre, o algoritmo precisa retornar as duas representações possíveis. Este detalhe, todavia, não chega a tornar o processo muito mais complicado. Somente se faz necessária a verificação dos dois fenômenos e o tratamento duplicado em casos como o do exemplo apresentado.

4.2.1 Novo modelo de conhecimento de mundo

Sabe-se que uma zero-anáfora não pode ocorrer se o correferente do elemento elidido é animado. Sendo assim, esta característica atribuída aos objetos representados pela linguagem é fundamental na escolha do tratamento a ser dado a uma ocorrência de apagamento verbal. Como visto na seção 3.3.3, esta informação pode ser extraída da ontologia.

Mas o eficiente modelo proposto por Saiz-Noeda e Palomar não é capaz representar alguns detalhes importantes para a abordagem que pretendo aqui. Já foi dito que o conceito de complementariedade, importante para o tratamento das elipses de *SV*, é baseado nas informações ontológicas dos constituintes da sentença. Para esclarecer melhor este ponto, considere novamente o exemplo (4.9d), repetido abaixo.

(4.14) João odeia [comer arroz],. Maria adora \emptyset_i .

Para verificar se o *SV* da primeira sentença é complementar ao *SV* da segunda, o algoritmo compara as informações ontológicas trazidas pelos seus componentes. Como o único componente presente no *SV* da segunda sentença é o verbo *adora*, suas características são comparadas às de *odeia*. Mas o que significa dizer que *adora* é complementar a *odeia*? Ou, formulando a pergunta de maneira mais geral: Por que *adora* pode ser colocado no lugar de *odeia* na segunda sentença do discurso (4.14)? A resposta é, o complemento de *adora* pode ser exatamente o mesmo que o de *odeia*. Logo, no momento de testar a complementariedade dos verbos, é preciso saber quais-são os complementos permitidos para cada um.

Por esta razão, propus uma pequena extensão do modelo de ontologia proposto por Saiz-Noeda e Palomar, no qual pode-se definir compatibilidade, não só entre sujeito ou complemento e verbo, mas entre dois verbos também.

Definição 26 (Compatibilidade entre verbos)

Sejam v_1 e v_2 duas folhas de O_v , denotando dois verbos, então $v_1 \in v_2$ se e somente se para todo n tal que $n \in_{ob} v_1$ também ocorre que $n \in_{ob} v_2$. ■

De acordo com a definição acima, dois verbos são compatíveis se possuírem o mesmo conjunto de objetos possíveis.

Da mesma forma que os verbos, pode ser necessário verificar a compatibilidade entre substantivos, como pode ser visto no exemplo (4.9f), repetido abaixo.

(4.15) João [odeia comer]₁ arroz. Maria, \emptyset ₁ feijão.

Para casos como este, propus a compatibilidade entre substantivos, definida abaixo.

Definição 27 (Compatibilidade entre substantivos)

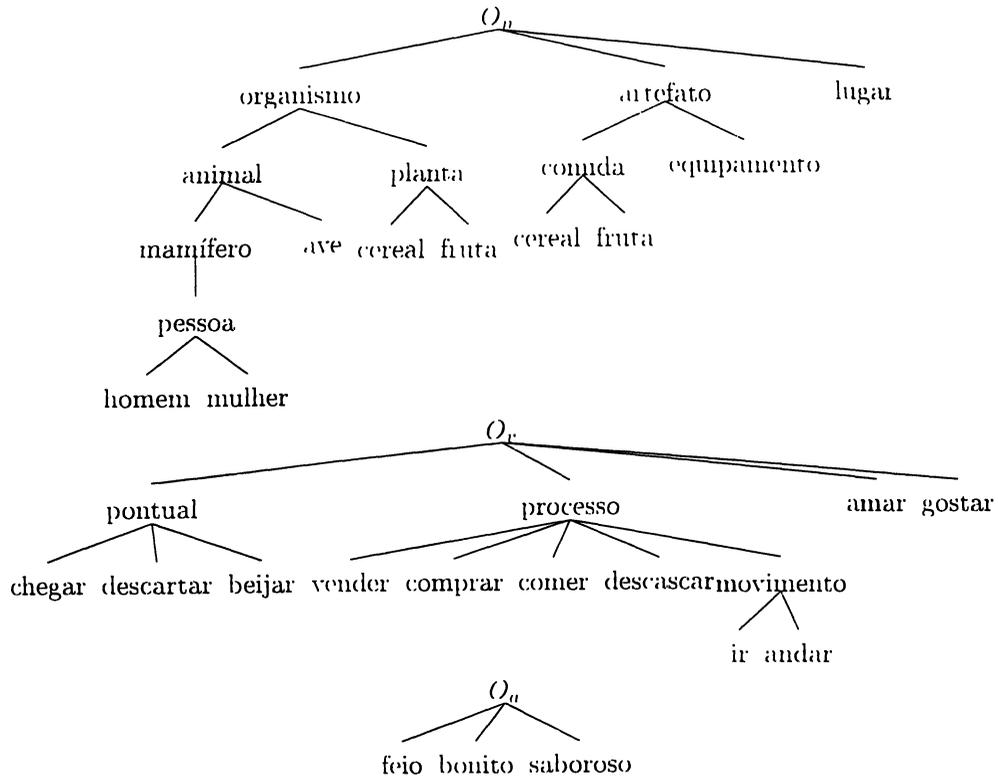
Sejam n_1 e n_2 folhas de O_n denotando substantivos, então $n_1 \in n_2$ se e somente se para todo v tal que $n_1 \in_{ob} v$ também ocorre que $n_2 \in_{ob} v$. ■

Nesta definição, dois substantivos são compatíveis quando são compatíveis como objetos com os mesmos verbos.

Propus também algumas pequenas mudanças na própria estrutura da árvore ontológica. Parte destas modificações pode ser vista na figura 4.1. A primeira mudança é a construção da árvore O_n . Esta classifica os adjetivos da linguagem, os quais estão relacionados entre si e com os substantivos, como apresentado nas definições acima.

As outras mudanças foram inseridas simplesmente para adaptar a árvore ao Português, aos constituintes léxicos e à gramática utilizada. Observe que eu não coloquei os constituintes nos nós folha das árvores. Na nova abordagem, as regras gramaticais trazem as informações ontológicas, classificando cada um dos constituintes. Isto também torna a classificação mais geral, pois desobriga a colocação das palavras como folhas das árvores. Permito, portanto, a utilização de qualquer nível da árvore para classificar um constituinte. Pode-se classificar uma palavra como ornitorrinco, por exemplo, como ‘mamífero’, ou como ‘animal’, ou como ‘organismo’.

Algumas das relações de compatibilidade propostas são dadas abaixo.

Figura 4.1: Nova ontologia (O_n , O_n e O_n)

$$R_{s_7} = \{ \begin{array}{l} (animal, chegar). \\ (animal, movimento), \\ (pessoa, vender). \\ (pessoa, comprar). \\ (animal, comer), \\ (pessoa, amar) \\ (objeto, feio) \end{array} \}$$

$$R_{ob} = \{ \begin{array}{l} (lugar, ir), \\ (comida, comer), \\ (O_n, gostar), \\ (arte fato, vender), \\ (lugar, vender). \\ (arte fato, comprar), \\ (lugar, comprar), \\ (fruta, descascar) \end{array} \}$$

$$R_{ad} = \{ \begin{array}{l} (O_n, \text{fcrto}). \\ (O_n, \text{bonuto}). \\ (\text{comida}, \text{saboroso}) \end{array} \}$$

Saiz-Noeda e Palomar não previram a necessidade de compatibilização entre substantivos e adjetivos. Estes últimos, colocados como complemento de verbos como ser e estar também precisam ser testados em caso de apagamento. Sendo assim, outras duas definições são necessárias.

Definição 28 (Compatibilidade entre sujeito e predicativo)

Seja a uma folha de O_a denotando um adjetivo, n uma folha de O_n denotando um substantivo. A_a o conjunto de ancestrais de a e A_n o conjunto de ancestrais de n , então $n \in_{ad} a$ sse $\exists (c_n, c_a) \in A_n \times A_a \mid (c_n, c_a) \in R_{ad}$. ■

Definição 29 (Compatibilidade entre adjetivos)

Sejam a_1 e a_2 folhas de O_a denotando adjetivos, então $a_1 \in a_2$ se e somente se para todo n tal que $n \in_{ad} a_1$ também ocorre que $n \in_{ad} a_2$ e para todo n tal que $n \in_{ad} a_2$ também ocorre que $n \in_{ad} a_1$. ■

Na realidade, as definições 26, 27, 28 e 29 são excessivamente restritivas. Em 4.14, por exemplo, é suficiente saber se o complemento do verbo da sentença anterior, neste caso comer arroz, pode figurar como complemento de adora. Não é necessário verificar se todos os complementos possíveis do verbo odiar são também aceitos como complemento de adorar. Contudo, o algoritmo de verificação de compatibilidade entre dois verbos, pela maneira que foi implementado, não tem acesso a qual é o complemento em cada situação em que o teste é feito. Sendo assim, para que existisse um mínimo de coerência entre as sentenças aceitas e as não aceitas, as regras foram definidas da forma em que se apresentam nas definições acima.

4.2.2 Zero-anáfora

Nesta seção, apresento o tratamento dado às zero-anáforas. Como já dito, é necessário produzir a palavra vazia na posição de objeto da sentença. Por esta razão, a Gramática 2 definida anteriormente precisa ser modificada.

Definição 30 (Modificação da Gramática 2)

$$(G2.12) \quad \begin{array}{c} SN \\ \left[\begin{array}{l} num:\square \\ sem:\square \end{array} \right] \end{array} \rightarrow \begin{array}{c} Z \\ \left[\begin{array}{l} sem:\square \end{array} \right] \end{array}$$

■

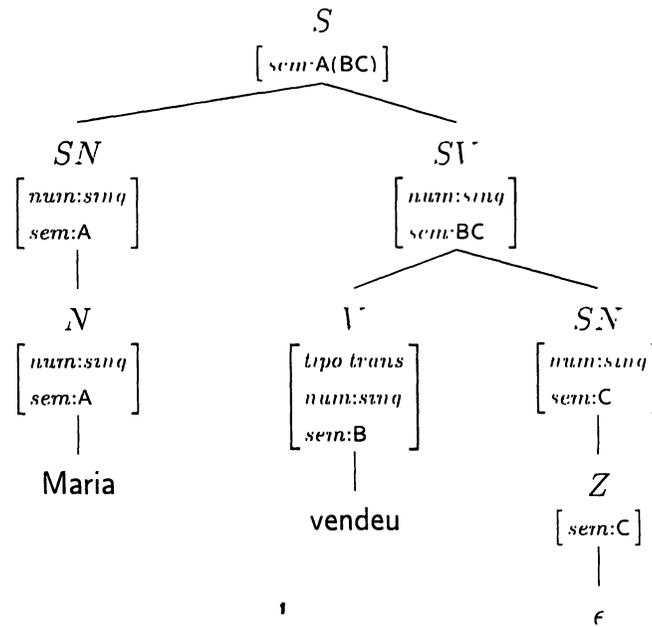


Figura 4.2: Árvore sintática: Maria vendeu. (zero-anáfora)

Apenas uma regra da gramática é modificada. A regra (G2.12), que antes aceitava a zero-anáfora apenas em posição de sujeito da sentença permite a produção de ϵ no lugar de um SN qualquer, tanto na posição de sujeito como na posição de objeto. Para ver como o procedimento funciona, considere o exemplo (4.12a). A representação semântica da primeira sentença é

$x_1 \ x_2$
$x_1 = ana$
$holo(x_2)$
$comprou(x_1, x_2)$

A árvore sintática gerada pela segunda sentença do discurso é dada na figura 4.2. Observe que a regra (G2.12) produz a palavra vazia na posição de objeto na segunda sentença. A segunda sentença é representada por A(BC). Realizando a conversão beta:

$$\begin{aligned}
 & \lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda P' \lambda x' \left(P' \lambda y \frac{}{vendeu(x', y)} \right) (\lambda P'' \alpha a_{zero}(P'' a)) \right) \\
 & \lambda P \left(\frac{x}{x = maria} \oplus (P.r) \right) \left(\lambda x' \left((\lambda P'' \alpha a_{zero}(P'' a)) \lambda y \frac{}{vendeu(x', y)} \right) \right) \\
 & \lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda x' \alpha a_{zero} \left(\lambda \tilde{y} \frac{}{vendeu(x', y)} a \right) \right)
 \end{aligned}$$

$$\begin{aligned}
& \lambda P \left(\begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline \end{array} \oplus (P.x) \right) \left(\lambda x' \alpha a_{zero} \begin{array}{|c|} \hline \\ \hline vendeu(x', a) \\ \hline \end{array} \right) \\
& \begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline \end{array} \oplus \left(\left(\lambda x' \alpha a_{zero} \begin{array}{|c|} \hline \\ \hline vendeu(x', a) \\ \hline \end{array} \right) x \right) \\
& \begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline \end{array} \oplus \alpha a_{zero} \begin{array}{|c|} \hline \\ \hline vendeu(x, a) \\ \hline \end{array} \\
& \alpha a_{zero} \begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline vendeu(x, a) \\ \hline \end{array}
\end{aligned}$$

A expressão que representa o discurso é dada pela combinação das representações das duas sentença. Esta é dada por

$$\alpha a_{zero} \begin{array}{|c|} \hline x_1 \ x_2 \ x_3 \\ \hline x_1 = ana \\ bolo(x_2) \\ comprou(x_1, x_2) \\ x_3 = maria \\ vendeu(x_3, a) \\ \hline \end{array}$$

A variável a só pode ser ligada a um referente de discurso que denote um objeto inanimado. Esta informação, advinda do conhecimento de mundo, precisa ser verificada pelo procedimento de escolha do correferente de a . Esta tarefa é realizada pelo algoritmo de resolução de ocorrências fóricas. Vale lembrar que este dá preferência ao referente no topo da pilha correspondente apenas se o elemento é aceito pelas restrições de gênero, número e contra-índice e consistência semântica.

Executando o algoritmo, obtém-se a expressão que representa o discurso completo. Esta é equivalente a apresentada anteriormente em (4.13a).

x_1, x_2, x_3
x_3, x_1
x_2
$x_1 = ana$ $bolo(x_2)$ $comprou(x_1, x_2)$ $x_3 = maria$ $vendeu(x_3, a)$ $a = x_2$

Sabe-se que esta não é a única seqüência de passos possível para a interpretação deste discurso. Uma outra representação deve pode ser retornada pelo procedimento. Na seção seguinte apresento o algoritmo capaz de realizar esta tarefa.

4.2.3 Eclipse de *SV*

O tratamento que proponho para a elipse de *SV* realiza uma cópia sintática dos elementos de uma sentença anteriormente proferida no discurso. Como visto nas seções anteriores, não são todas as ocorrências de apagamento que devem ser consideradas elipses de *SV*. No momento em que um fenômeno destes é detectado, é preciso realizar uma checagem utilizando informações provenientes de sentenças anteriores. Estas mesmas informações servem para resolver a ocorrência de apagamento, reconstruindo o *SV* parcialmente apagado.

Após a reconstrução, o procedimento pode montar a representação do discurso em λ -DRT, realizar a conversão beta e retornar a representação semântica. Logo, a reconstrução do *SV* deve ser feita antes da conversão beta, juntamente com o reconhecedor sintático.

A definição 25 não explicita qual é a distância máxima entre uma elipse de *SV* e o *SV* anterior a ser parcialmente copiado. Logo, qualquer *SV* de uma sentença anterior é candidato a cópia. O reconhecedor precisa, portanto, acumular as informações sintáticas de todas as sentenças do discurso até o final do reconhecimento sintático.

Mas quais são as informações de cada sentença que precisam ser acumuladas? De acordo com a definição, o procedimento deve ser capaz de comparar os *SVs* das sentenças e verificar sua complementariedade. Lembrando os discursos do exemplo (4.9), para dar um tratamento completo à complementariedade, várias informações do nível pragmático precisariam ser modeladas. O conhecimento de mundo cresceria de maneira espantosa, além da necessidade de mecanismos de inferência aplicados a formalismos lógicos diversos.

Contudo, a definição deixa claro que as estruturas hierárquicas do *SV* incompleto e do seu correspondente complementar são iguais, pois todos os elementos apagados podem ser copiados deste *SV*. Além disso, os elementos que permanecem no *SV* incompleto devem

ser compatíveis, de acordo com as definições de compatibilidade da seção 4.2.1, com os elementos que estão na mesma posição do SV complementar. Claro, pois se os dois verbos não admitirem os mesmos tipos de objeto, por exemplo, o complemento verbal de um dos SV 's não pode ser copiado para o lugar do complemento apagado do outro. De forma análoga, se a complementaridade é dada pelos objetos dos verbos, o verbo de um SV não deve ser copiado sem que o complemento verbal do SV incompleto seja compatível.

Conclui-se, portanto, que duas informações são cruciais: a estrutura sintática do SV e as informações ontológicas dos elementos presentes nesta estrutura. A análise sintática deve ser feita de forma a passar estas informações de uma sentença para a outra. Com estas informações, um novo procedimento verifica se os SV 's são complementares e coloca as informações ausentes no SV incompleto. Ao final, o reconhecedor sintático retorna a representação semântica em λ -DRT.

Abaixo, apresento a nova gramática responsável por fazer o reconhecimento sintático de discursos. Agora, ela é capaz de reconhecer discursos contendo os dois tipos de objeto nulo a serem tratados. Além disso, ela incorpora o procedimento de resolução das elipses de SV .

Definição 31 (Gramática 3)

$$(G3.1) \quad D \rightarrow S \\ \left[\begin{array}{l} in:\square \\ out:[\mathcal{R}(\square,\square)]_D \end{array} \right] \quad \left[\begin{array}{l} out:\square \end{array} \right]$$

$$(G3.2) \quad D \rightarrow S \quad C \quad D \\ \left[\begin{array}{l} in:\square \\ out:[\square,\mathcal{R}(\square,\square),\square]_D \end{array} \right] \quad \left[\begin{array}{l} out:\square \end{array} \right] \quad \left[\begin{array}{l} in:\mathcal{R}(\square,\square) \\ out:\square \end{array} \right]$$

$$(G3.3) \quad S \rightarrow SN \quad SSV \\ \left[\begin{array}{l} out:[\square,\square]_S \end{array} \right] \quad \left[\begin{array}{l} num:\square \\ out:\square \end{array} \right] \quad \left[\begin{array}{l} num:\square \\ out:\square \end{array} \right]$$

$$(G3.4) \quad SN \rightarrow Det \quad N \\ \left[\begin{array}{l} num:\square \\ out:[\square,\square]_{SN} \end{array} \right] \quad \left[\begin{array}{l} num:\square \\ out:\square \end{array} \right] \quad \left[\begin{array}{l} num:\square \\ out:\square \end{array} \right]$$

$$(G3.5) \quad SN \rightarrow N \\ \left[\begin{array}{l} num:\square \\ out:[[\emptyset,\emptyset]_{Det},\square]_{SN} \end{array} \right] \quad \left[\begin{array}{l} num:\square \\ out:\square \end{array} \right]$$

$$(G3.6) \quad SN \rightarrow Z \\ \left[\begin{array}{l} out:[[\emptyset,\emptyset]_{Det},\square]_{SN} \end{array} \right] \quad \left[\begin{array}{l} out:\square \end{array} \right]$$

$$(G3.7) \quad SSV^{\bar{r}} \rightarrow ProS \quad Mod \quad SV^{\bar{r}}$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2},\bar{2},\bar{4}]_{SSV^{\bar{r}}} \end{bmatrix} \quad \begin{bmatrix} out:\bar{2} \end{bmatrix} \quad \begin{bmatrix} out:\bar{3} \end{bmatrix} \quad \begin{bmatrix} num:\bar{1} \\ out:\bar{1} \end{bmatrix}$$

$$(G3.8) \quad SV^{\bar{r}} \rightarrow V^{\bar{r}}$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2}]_{SV^{\bar{r}}} \end{bmatrix} \quad \begin{bmatrix} cat:intr \\ num:\bar{1} \\ out:\bar{2} \end{bmatrix}$$

$$(G3.9) \quad SV \rightarrow V^{\bar{r}} \quad SN$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2},\bar{2}]_{SV} \end{bmatrix} \quad \begin{bmatrix} cat:trans \\ num:\bar{1} \\ out:\bar{2} \end{bmatrix} \quad \begin{bmatrix} out:\bar{2} \end{bmatrix}$$

$$(G3.10) \quad SV \rightarrow V^{\bar{r}} \quad SVI$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2},\bar{2}]_{SV} \end{bmatrix} \quad \begin{bmatrix} cat:trans \\ num:\bar{1} \\ out:\bar{2} \end{bmatrix} \quad \begin{bmatrix} num:\bar{1} \\ out:\bar{2} \end{bmatrix}$$

$$(G3.11) \quad SV \rightarrow V^{\bar{r}} \quad A$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2},\bar{2}]_{SV} \end{bmatrix} \quad \begin{bmatrix} cat:pred \\ num:\bar{1} \\ out:\bar{2} \end{bmatrix} \quad \begin{bmatrix} num:\bar{1} \\ out:\bar{2} \end{bmatrix}$$

$$(G3.12) \quad SVI \rightarrow V^{\bar{r}} \quad SN$$

$$\begin{bmatrix} num:\bar{1} \\ out:[\bar{2},\bar{2}]_{SVI} \end{bmatrix} \quad \begin{bmatrix} cat:mf \\ num:\bar{1} \\ out:\bar{2} \end{bmatrix} \quad \begin{bmatrix} num:\bar{1} \\ out:\bar{2} \end{bmatrix}$$

$$(G3.13) \quad SV \rightarrow \epsilon$$

$$\begin{bmatrix} out:\mathcal{V}_{SV} \end{bmatrix}$$

$$(G3.14) \quad SN \rightarrow \epsilon$$

$$\begin{bmatrix} out:\mathcal{V}_{SN} \end{bmatrix}$$

$$(G3.15) \quad SVI \rightarrow \epsilon$$

$$\begin{bmatrix} out:\mathcal{V}_{SVI} \end{bmatrix}$$

Entradas léxicas:

$$(G3.15) \quad Det \rightarrow um$$

$$\begin{bmatrix} num:sing \\ out: \left[\left(\emptyset, \lambda P \lambda Q \left(\left(\begin{bmatrix} .r \\ \square \end{bmatrix} -Pr \right) \oplus Q.r \right) \right) \right]_{Det} \end{bmatrix}$$

(G3.16) \mathcal{N} \rightarrow Ana

$$\left[\begin{array}{l} \text{num:sing} \\ \text{out:} \left[\left(\text{mulher}.\lambda P \left(\begin{array}{c} e \\ x = \text{ana} \end{array} \right) \dot{\vdash} (Pr) \right) \right] \end{array} \right]_{\mathcal{N}}$$

(G3.17) \mathcal{N} \rightarrow arroz

$$\left[\begin{array}{l} \text{num:sing} \\ \text{out:} \left[\left(\text{cereal}.\lambda P \left(\begin{array}{c} x \\ \text{arroz}(x) \end{array} \right) \dot{\vdash} (Pr) \right) \right] \end{array} \right]_{\mathcal{N}}$$

(G3.18) \mathcal{V}^* \rightarrow ama

$$\left[\begin{array}{l} \text{cat:trans} \\ \text{num:sing} \\ \text{out:} \left[\left(\text{amar}.\lambda P \lambda x \left(P \lambda y \begin{array}{c} \\ \text{ama}(x, y) \end{array} \right) \right) \right] \end{array} \right]_{\mathcal{V}^*}$$

(G3.19) \mathcal{V}^* \rightarrow comer

$$\left[\begin{array}{l} \text{cat:mf} \\ \text{num:sing} \\ \text{out:} \left[\left(\text{comer}.\lambda P \lambda Q \left(P \lambda x \begin{array}{c} e \\ e : \text{comer}(x) \end{array} \right) \oplus (Qe) \right) \right] \end{array} \right]_{\mathcal{V}^*}$$

(G3.20) \mathcal{V}^* \rightarrow ϵ

$$\left[\text{out:} \mathcal{V}_{\mathcal{V}^*} \right]$$

(G3.21) \mathcal{A} \rightarrow bonito

$$\left[\begin{array}{l} \text{num:sing} \\ \text{out:} \left[\left(\text{bonito}.\lambda x \begin{array}{c} \\ \text{bonito}(x) \end{array} \right) \right] \end{array} \right]_{\mathcal{A}}$$

(G3.22) \mathcal{C} \rightarrow .

$$\left[\text{out:} \{ (\emptyset.\lambda P \lambda Q (P \oplus Q)) \}_{\mathcal{C}} \right]$$

(G3.23) \mathcal{Z} \rightarrow ϵ

$$\left[\text{out:} \{ (\emptyset.\lambda P \alpha_{\text{zero}}(Px)) \}_{\mathcal{Z}} \right]$$

(G3.24) $ProS$ \rightarrow também

$$\left[\text{out:} \{ (\emptyset.\emptyset) \}_{ProS} \right]$$

(G3.25) $ProS$ \rightarrow ϵ

$$\left[\text{out:} \{ (\emptyset.\emptyset) \}_{ProS} \right]$$

$$(G3.26) \quad \text{Mod} \rightarrow \text{n\~{a}o}$$

$$\left[\text{out}: \left[\left(\emptyset, \lambda P \lambda x. \begin{array}{|c|} \hline \\ \hline \neg(P.x) \\ \hline \end{array} \right) \right]_{\text{Mod}} \right]$$

$$(G3.27) \quad \text{Mod} \rightarrow \epsilon$$

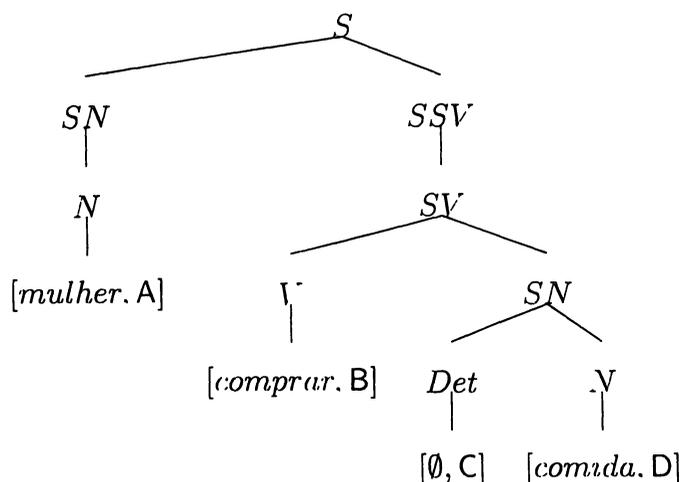
$$\left[\text{out}: [(\emptyset, \emptyset)_{\text{Mod}}] \right]$$

■

O ponto inicial desta gramática é novamente o discurso (D). Ou seja, as regras de produção iniciam com (G3.1) ou (G3.2). Contudo, uma diferença bastante visível entre a Gramática 2 e a Gramática 3 é a inclusão dos atributos in e out nas duas primeiras regras, assim como o atributo out nas demais. O atributo in nada mais é do que aquele que carrega as informações necessárias para a resolução da elipse de SV na sentença atual. O atributo out é aquele que leva as informações para as próximas sentenças. Esta representação aproveita um sistema de listas de listas, equivalentes às estruturas de árvore. Como ilustração, considere o exemplo de uma lista que representa o discurso Ana comprou um bolo, reconhecido pela gramática.

$$X = [[[mulher, A]_N]_{SN}, [[[comprar, B]_V, [[\emptyset, C]_{Det}, [comida, D]_N]_{SN}]_{SV}]_{SSV}]_S$$

Por brevidade, os símbolos A, B, C e D substituem as expressões lambda que representam semanticamente cada um dos constituintes. A lista de listas X , acima, também pode ser encarada como uma estrutura de árvore, como abaixo.



Observe que além da estrutura hierárquica de todo o discurso, in e out também carregam as informações ontológicas e a representação semântica de todos os constituintes. Em alguns casos, porém, como nas regras (G3.13), (G3.14), (G3.15) e (G3.20), estas estruturas são instanciadas com variáveis livres. Estas, simbolizadas por \mathcal{V} , devem ser preenchidas com informações provenientes de uma sentença anteriormente proferida no discurso. Sendo assim, o reconhecedor sintático também produz ϵ nos casos de elipse.

Mas a elipse é tratada diferentemente da zero-anáfora. Esta introduz uma representação utilizando o operador α , como pode ser visto na regra (G3.23). Logo, a zero-anáfora é resolvida por um procedimento posterior, no nível semântico, enquanto que a elipse de SV é resolvida no nível sintático.

A estrutura de lista de listas apresentada acima é a que o reconhecedor sintático retornaria para o procedimento imediatamente superior, que realiza a conversão beta. Para que esta representação possa ser utilizada, um novo passo foi introduzido na computação do discurso. Uma função de tradução da representação hierárquica para a λ -DRT.

Definição 32 (Algoritmo de tradução – \mathcal{T})

1. $\mathcal{T}([a_1, \dots, \emptyset, \dots, a_n]_\phi) ::= \mathcal{T}([a_1, \dots, a_n])$.
2. $\mathcal{T}([a_1, \dots, a_n]_\phi) ::= \mathcal{T}([a_n, \dots, a_1])$, onde $a_i \neq \emptyset$.
3. $\mathcal{T}([\])$::= \emptyset .
4. $\mathcal{T}([a_1, \dots, a_n]) ::= \mathcal{T}(a_n) \cdots \mathcal{T}(a_1)$, onde $a_i \neq \emptyset$.
5. $\mathcal{T}((c, t)) ::= t$. ■

Se a função for aplicada à estrutura X do exemplo, ela agirá recursivamente retornando as representações lambda já aplicadas umas às outras na ordem correta, construindo a representação do discurso completo.

$$\begin{aligned}
& \mathcal{T}([[[[mulher, A]_N]_{SN}, [[comprar, B]_V, [[\emptyset, C]_{Det}, [comida, D]_N]_{SN}]_{SV}]_{SSV}]_S) = \\
& \mathcal{T}([[[[comprar, B]_V, [[\emptyset, C]_{Det}, [comida, D]_N]_{SN}]_{SV}]_{SSV}, [[mulher, A]_N]_{SN}]) = \\
& \mathcal{T}([[[mulher, A]_N]_{SN}] \mathcal{T}([[[comprar, B]_V, [[\emptyset, C]_{Det}, [comida, D]_N]_{SN}]_{SV}]_{SSV})) = \\
& \mathcal{T}([mulher, A]_N) \mathcal{T}([comprar, B]_V, [[\emptyset, C]_{Det}, [comida, D]_N]_{SN})_{SV} = \\
& A\mathcal{T}([[[\emptyset, C]_{Det}, [comida, D]_N]_{SN}, [comprar, B]_V]) = \\
& A(\mathcal{T}([comprar, B]_V) \mathcal{T}([[\emptyset, C]_{Det}, [comida, D]_N]_{SN}))) = \\
& A(B(\mathcal{T}([comida, D]_N, [\emptyset, C]_{Det}))) = \\
& A(B(\mathcal{T}([\emptyset, C]_{Det}) \mathcal{T}([comida, D]_N))) = \\
& A(B(CD))
\end{aligned}$$

Portanto, a despeito das modificações introduzidas em relação à Gramática 2, os procedimentos que tratam dos formalismos lógicos vistos até aqui não precisam sofrer nenhuma alteração. Logo, tudo que foi apresentado, inclusive o tratamento da zero-anáfora, pode ser reiterado para a Gramática 3.

Continuando a analisar a Gramática 3, pode-se observar que as regras (G3.1) e (G3.2) implementam um esquema de passagem das informações armazenadas nos atributos *out* de uma sentença para outra. Com estas informações é possível, no nível da sentença, resolver as elipses de SV . A resolução é implementada pela função \mathcal{R} . Esta função

recebe a representação da sentença atual. verifica se existe uma elipse de SV^* validada, comparando-a com os SV^* s das sentenças anteriores, e retorna o SV^* resolvido. As demais regras da gramática são construídas de forma a aceitar o apagamento parcial dos SV^* s sem deixar de reconhecer outros tipo de fenômenos fóricos, não só em posição de complemento verbal como na posição de agente da sentença.

Abaixo, apresento a função \mathcal{R} que implementa a verificação e a resolução das elipses de SV^* .

Definição 33 (Algoritmo de solução de elipses de SV^* – \mathcal{R})

Sejam A e B duas listas de listas, então

$$\mathcal{R}([A_{SN}, A_{SSV^*}]_S, [B_{SN}, B_{SSV^*}]_S) ::= [B_{SN}, \mathcal{C}(A_{SSV^*}, B_{SSV^*})]_S \quad \blacksquare$$

Ou seja, \mathcal{R} retorna como resultado uma sentença cujo primeiro constituinte é o SN da segunda sentença. O segundo constituinte resulta da aplicação do algoritmo \mathcal{C} aos dois SSV^* . O super sintagma verbal (SSV^*) engloba todos os constituintes importantes na verificação da complementariedade entre os sintagmas: o prossintagma verbal ($ProS$), o modificador do verbo (Mod) e o SV^* . A definição de \mathcal{C} , por sua vez, nada mais é do que a implementação da definição de complementariedade. Ela é apresentada abaixo.

Definição 34 (Algoritmo de complementariedade – \mathcal{C})

Sejam $A = [a_1, \dots, a_n]_\phi$ e $B = [b_1, \dots, b_n]_\phi$ duas listas de listas e \mathcal{V} uma variável livre, então

1. $\mathcal{C}(A, B) ::= [C(a_1, b_1)]_\phi + [\bar{\mathcal{C}}([a_2, \dots, a_n], [b_2, \dots, b_n])]_\phi$.
2. $\mathcal{C}(A, B) ::= [\bar{\mathcal{C}}(a_1, b_1)]_\phi + [C([a_2, \dots, a_n], [b_2, \dots, b_n])]_\phi$.
3. $\mathcal{C}([(c_a, s_a)]_N, [(c_b, s_b)]_N) ::= [(c_b, s_b)]_N$
se $c_a \in c_b$ e $s_a \neq s_b$.
4. $\mathcal{C}([(c_a, s_a)]_{\mathcal{V}}, [(c_b, s_b)]_{\mathcal{V}}) ::= [(c_b, s_b)]_{\mathcal{V}}$
se $c_a \in c_b$ e $s_a \neq s_b$.
5. $\mathcal{C}([(c_a, s_a)]_A, [(c_b, s_b)]_A) ::= [(c_b, s_b)]_A$
se $c_a \in c_b$ e $s_a \neq s_b$.
6. $\mathcal{C}([(c_a, s_a)]_\phi, [(c_b, s_b)]_\phi) ::= [(c_b, s_b)]_\phi$
se $\neg \bar{\mathcal{C}}([(c_a, s_a)]_\phi, [(c_b, s_b)]_\phi)$.
7. $\bar{\mathcal{C}}(A, B) ::= [\bar{\mathcal{C}}(a_1, b_1)]_\phi + [\bar{\mathcal{C}}([a_2, \dots, a_n], [b_2, \dots, b_n])]_\phi$.
8. $\bar{\mathcal{C}}([(c_a, s_a), (c_b, s_b)]_N) ::= [(c_b, s_b)]_N$
se $s_a = s_b$.

9. $\bar{\mathcal{C}}([(c_a, s_a), (c_b, s_b)]_V) ::= [(c_b, s_b)]_V$
se $s_a = s_b$.
10. $\bar{\mathcal{C}}([(c_a, s_a), (c_b, s_b)]_A) ::= [(c_b, s_b)]_A$
se $s_a = s_b$.
11. $\bar{\mathcal{C}}([(c_a, s_a)]_{ProS}, [(c_b, s_b)]_{ProS}) ::= [(c_b, s_b)]_{ProS}$
se $s_a = s_b$.
12. $\bar{\mathcal{C}}([(c_a, s_a)]_{Det}, [(c_b, s_b)]_{Det}) ::= [(c_b, s_b)]_{Det}$
se $s_a = s_b$ ou $(s_a = \emptyset)$ ou $(s_b = \emptyset)$.
13. $\bar{\mathcal{C}}([(c_a, s_a)]_{Mod}, [(c_b, s_b)]_{Mod}) ::= [(c_b, s_b)]_{Mod}$
se $s_a = s_b$ ou $(s_a = \emptyset \wedge s_b = \text{sim})$ ou $(s_a = \text{sim} \wedge s_b = \emptyset)$.
14. $\bar{\mathcal{C}}(A, \mathcal{V}_\phi) ::= A$. ■

Intuitivamente, \mathcal{C} representa o complementar, enquanto que $\bar{\mathcal{C}}$ é o não complementar. Segue da definição acima, que $\mathcal{C}(A, B)$ verifica se B é complementar a A e retorna uma nova estrutura idêntica a B , a não ser pelas variáveis livres (\mathcal{V}), que são copiadas de A . Ou seja, se B , a menos das variáveis livres, é complementar a A , então \mathcal{C} copia o que falta em B de A . A nova estrutura é formada recursivamente concatenando as sub-listas (operador $+$). Os itens 1, 2 e 7 da definição 34 definem as regras gerais de complementariedade e não complementariedade. Enquadram-se nestas regras as estruturas SSV , SV , SN e, no caso de \mathcal{C} , também $ProS$ e Det . Observe que uma estrutura é complementar a outra se pelo menos uma das sub-estruturas é complementar. Por outro lado, ela é não complementar se todas as sub-estruturas forem não complementares.

As demais regras tratam os casos especiais de cada sub-estrutura que representa uma palavra na linguagem. Fato verificado através das classes ontológicas (c_a e c_b) e das representações semânticas (s_a e s_b) das palavras.

Duas estruturas N , V ou A (adjetivo) são complementares se são compatíveis e não são o mesmo constituinte. Todavia, duas estruturas N , V ou A são não complementares se forem exatamente o mesmo constituinte. As demais estruturas $ProS$, Mod e Det são complementares se não forem não complementares.

O determinante, Det , se estiver presente, precisa ser igual para ser considerado não complementar.

As estruturas Mod representam o modificador do verbo. Se uma delas estiver presente, ela é não complementar à outra se forem idênticas. Se uma das estruturas Mod representar a palavra *sim*, a outra não precisa estar presente, como no exemplo abaixo.

- (4.16) (a) João não [foi dormir]_i. Maria sim \emptyset _i.
(b) *João [foi dormir]_j. Maria sim \emptyset _j.

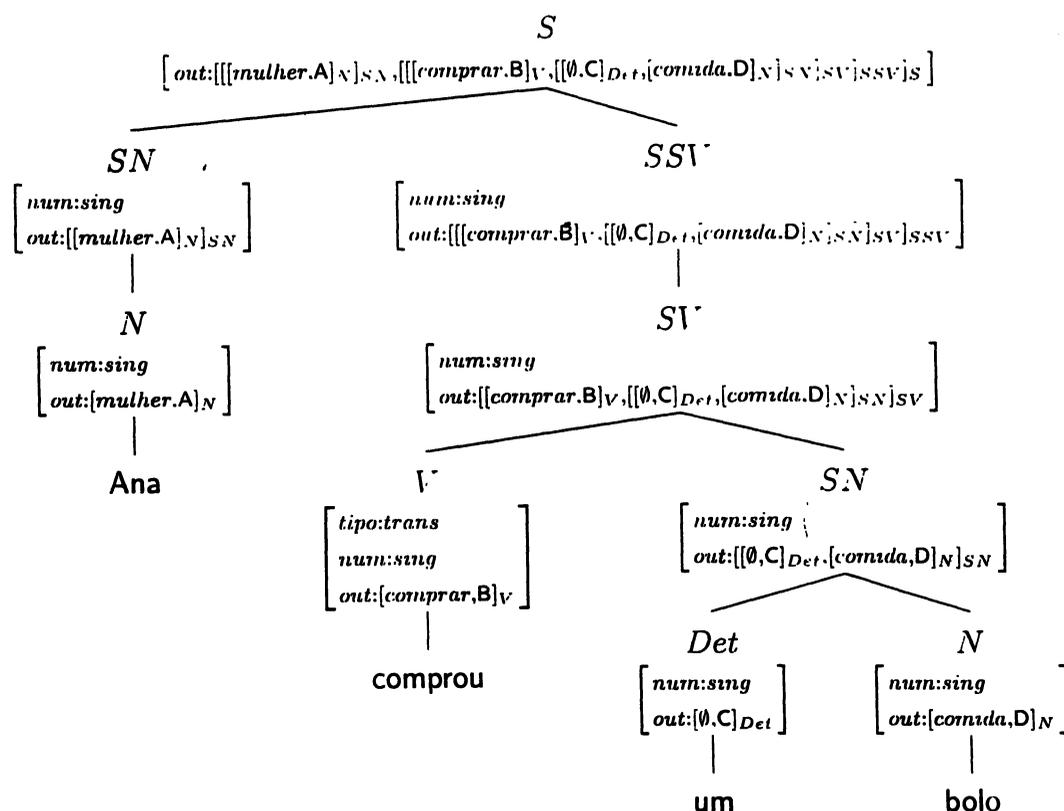


Figura 4.3: Árvore sintática: Ana comprou um bolo. (elipse de SV)

Observe que (4.16a) deve ser considerado complementar, o mesmo não ocorrendo com (4.16b). Isto se deve ao modificador *sim* ser implícito na primeira sentença de (4.16b).

As estruturas *ProS* representam os prossintagmas da linguagem. A Gramática 3 apresenta um único prossintagma, também. Dois prossintagmas precisam estar presentes e serem iguais para serem considerados não complementares. Esta regra é responsável por considerar os dois *SSV*s da sentença do exemplo abaixo complementares.

(4.17) João [foi dormir]_i. Maria também \emptyset _i.

Como o prossintagma *também* está presente em apenas uma delas, temos uma ocorrência de elipse de *SV*.

Finalmente, se uma das estruturas for uma variável livre, fato representado por \mathcal{V}_ϕ na regra 14, então as duas estruturas são não complementares. O resultado retornado pelo algoritmo é a primeira delas. Esta é a regra que implementa a cópia dos elementos apagados.

Como ilustração do funcionamento do procedimento descrito, considere novamente (4.12b), repetido aqui.

(4.18) Ana comprou [um bolo]_i. Maria vendeu \emptyset _i.

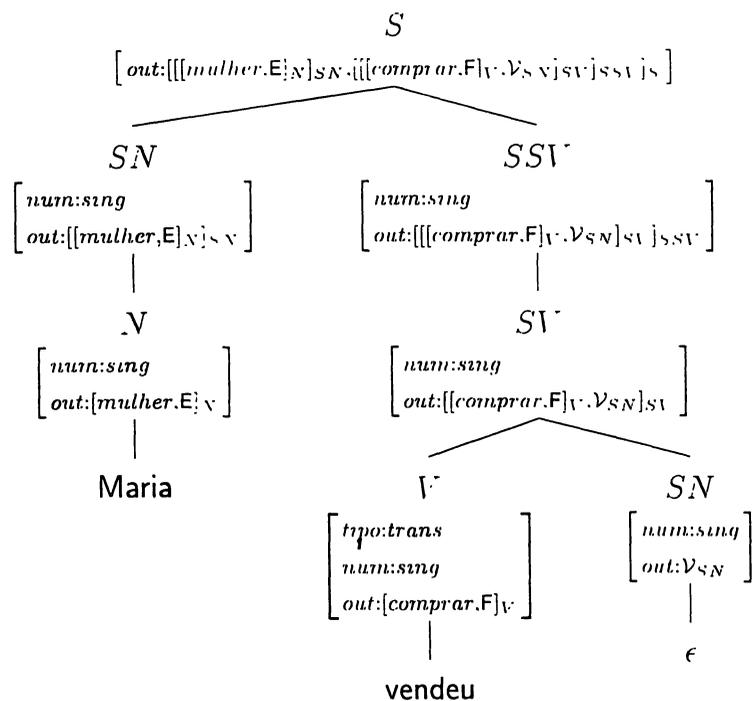


Figura 4.4: Árvore sintática: Maria vendeu. (transitivo)

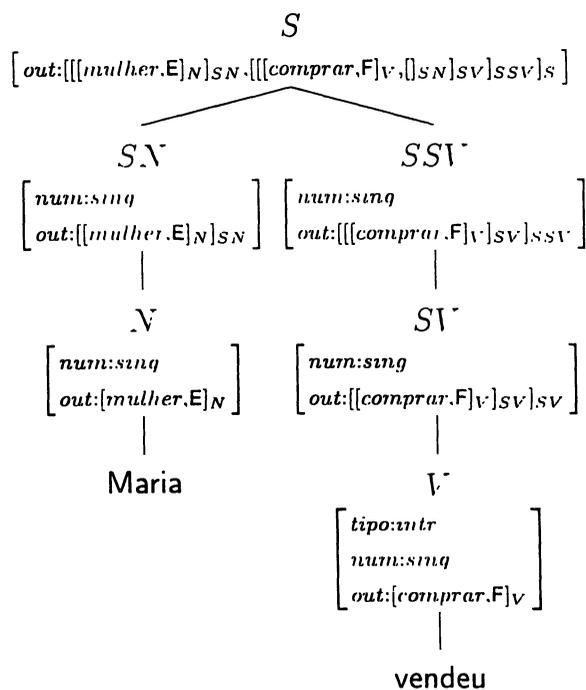


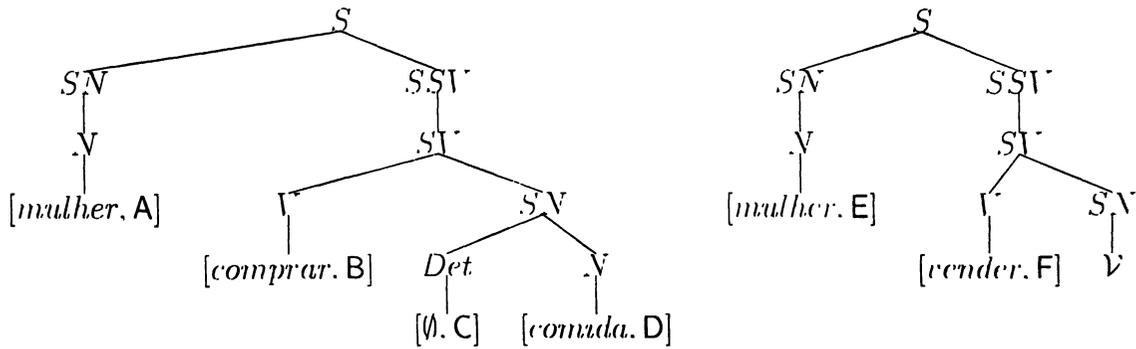
Figura 4.5: Árvore sintática: Maria vendeu. (intransitivo)

As árvores sintáticas das sentenças são dadas nas figuras 4.3 e 4.4. Aqui, A, B, C e D substituem as representações semânticas dadas em λ -DRT.

A árvore sintática da segunda sentença é apenas uma das árvores que podem ser geradas à partir da Gramática 3 definida acima. Esta é a árvore que permite a representação deste discurso como contendo uma eclipse de SV . Contudo, existem outras duas árvores possíveis. Uma delas é equivalente à apresentada na seção 4.2.2, que retorna a interpretação da sentença como se nela existisse uma zero-anáfora. A outra interpreta o verbo *vendeu* como intransitivo. Como ilustração, coloco-a na figura 4.5.

Esta ambigüidade, garante que o algoritmo retorne todas as interpretações possíveis para o discurso.

Continuando na computação do exemplo, o reconhecedor retorna a representação da sentença através do atributo *out*, na forma de lista de listas. Abaixo, mostro a representação equivalente, no formato de árvore.



Ainda dentro do reconhecedor sintático, na regra (G3.1), o algoritmo definido em 34 verifica se os SSV 's são complementares, construindo a nova representação do SSV do discurso.

$$C([[(comprar, B)]_V, [[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}]_{SV}]_{SSV}, [[(vender, F)]_V, \mathcal{V}_{SN}]_{SV}]_{SSV}$$

$$[C([[(comprar, B)]_V, [[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}]_{SV}, [[(vender, F)]_V, \mathcal{V}_{SN}]_{SV})]_{SSV} \quad [1]^1$$

$$[[C([[(comprar, B)]_V, [(vender, F)]_V])]_{SV} +$$

$$[\bar{C}([[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}, \mathcal{V}_{SN})]_{SV}]_{SSV} \quad [1]$$

$$[[[(vender, F)]_V]_{SV} + [\bar{C}([[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}, \mathcal{V}_{SN})]_{SV}]_{SSV} \quad [4]$$

$$[[[(vender, F)]_V]_{SV} + [[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}]_{SV}]_{SSV} \quad [14]$$

$$[[[(vender, F)]_V, [[(\emptyset, C)]_{Det}, [(comida, D)]_N]_{SN}]_{SV}]_{SSV}$$

A gramática retorna a representação do novo SSV e combina-a com a do restante do

¹Os números entre colchetes referem-se às regras do algoritmo de complementariedade (def. 34, pg. 67) aplicadas em cada passagem.

discurso.

$$[[[mulher. E]_N]_{SN}, [[[vender. F]_V, [[(\emptyset, C)]_{Det}, [(comida. D)]_N]_{SN}]_{SV}]_{SS}]_S$$

Esta representação é retornada para o reconhecedor sintático. Em seguida, é a vez do procedimento de tradução extrair a representação semântica do discurso e retorná-la como uma expressão lambda. Esta é dada por E(F(CD)). Abaixo, a conversão beta é realizada.

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda P' \lambda x' \left(P' \lambda y \frac{}{vendeu(x', y)} \right) \left(\lambda P'' \lambda Q \left(\left(\frac{x''}{} \oplus P'' x'' \right) \oplus Q x'' \right) \lambda x''' \frac{}{bolo(x''')} \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda P' \lambda x' \left(P' \lambda y \frac{}{vendeu(x', y)} \right) \left(\lambda Q \left(\left(\frac{x''}{} \oplus \lambda x''' \frac{}{bolo(x''')} x'' \right) \oplus Q x'' \right) \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda P' \lambda x' \left(P' \lambda y \frac{}{vendeu(x', y)} \right) \left(\lambda Q \left(\left(\frac{x''}{} \oplus \frac{}{bolo(x'')} \right) \oplus Q x'' \right) \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda P' \lambda x' \left(P' \lambda y \frac{}{vendeu(x', y)} \right) \left(\lambda Q \left(\frac{x''}{bolo(x'')} \oplus Q x'' \right) \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda x' \left(\lambda Q \left(\frac{x''}{bolo(x'')} \oplus Q x'' \right) \lambda y \frac{}{vendeu(x', y)} \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda x' \left(\frac{x''}{bolo(x'')} \oplus \lambda y \frac{}{vendeu(x', y)} x'' \right) \right)$$

$$\lambda P \left(\frac{x}{x = maria} \oplus (Px) \right) \left(\lambda x' \left(\frac{x''}{bolo(x'')} \oplus \frac{}{vendeu(x', x'')} \right) \right)$$

$$\lambda P \left(\begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline \end{array} \oplus (P.x) \right) \left(\begin{array}{|c|} \hline x'' \\ \hline \lambda x' \begin{array}{|c|} \hline bolo(x'') \\ \hline vendeu(x', x'') \\ \hline \end{array} \\ \hline \end{array} \right)$$

$$\begin{array}{|c|} \hline x \\ \hline x = maria \\ \hline \end{array} \ni \begin{array}{|c|} \hline x'' \\ \hline bolo(x'') \\ \hline vendeu(x, x'') \\ \hline \end{array}$$

$$\begin{array}{|c|} \hline x \ x'' \\ \hline x = maria \\ \hline bolo(x'') \\ \hline vendeu(x, x'') \\ \hline \end{array}$$

Desta forma, o procedimento descrito até aqui pode dar o tratamento pretendido ao objeto nulo no Português do Brasil. O procedimento completo é apresentado abaixo.

Definição 35 (Algoritmo de representação de discursos)

Seja D um discurso, a lista $[r_1, \dots, r_n]$, onde r_i denota cada uma das possíveis representações de D é dado por $(\mathcal{F} \circ \beta \circ \mathcal{T} \circ \mathcal{S})(D)$. ■

Ou seja, um discurso (D) pode possuir várias representações (r_i). Todas elas são resultado da aplicação da composição das funções que denotam os algoritmos de *focusing* (\mathcal{F}), da conversão beta (β), da tradução (\mathcal{T}) e do reconhecedor sintático (\mathcal{S}) ao discurso. As várias representações são possíveis devido as várias respostas que a aplicação de \mathcal{S} pode retornar. Isto significa que em o reconhecedor sintático retornando três árvores, por exemplo, cada uma delas será traduzida por \mathcal{T} . Se a tradução obtiver sucesso, o resultado sofrerá uma conversão beta. O resultado deste passo será passado para \mathcal{F} que resolve as ocorrências fóricas se existirem e retorna a representação final.

A título de ilustração, observe o funcionamento do algoritmo aplicado a dois exemplos um pouco maiores. Estes são dois dos exemplos que motivaram a criação da Gramática 3, da função de tradução e do algoritmo de complementariedade.

(4.19) João odeia [comer arroz]_i. Maria também \emptyset_i .

(4.20) João [odeia comer]_j arroz. Maria, \emptyset_j feijão.

A árvore sintática das duas sentenças do exemplo (4.19) são dadas nas figuras 4.6 e 4.7. As representações retornadas pela gramática em forma de árvore são:

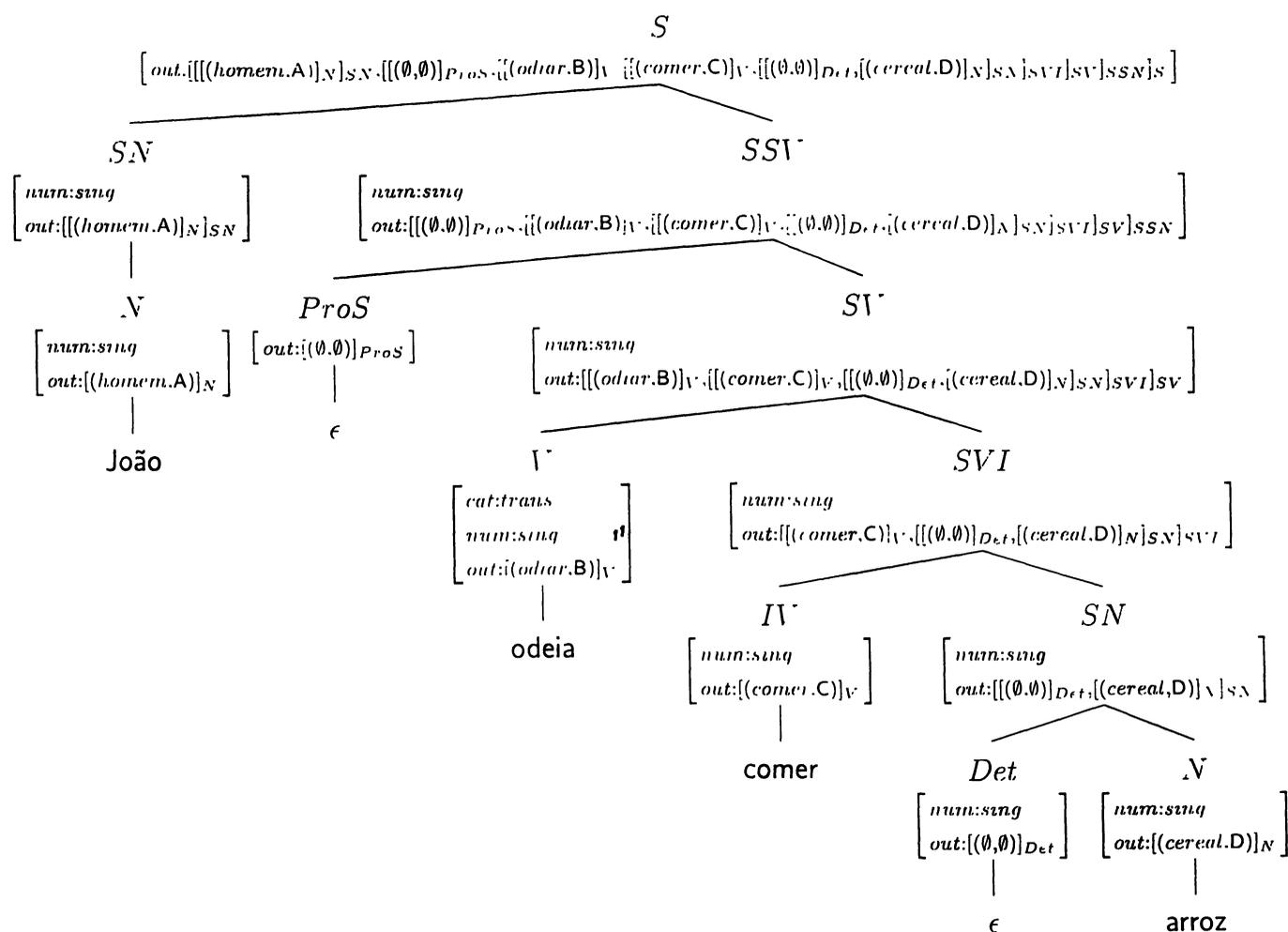


Figura 4.6: Árvore sintática: João odeia comer arroz.

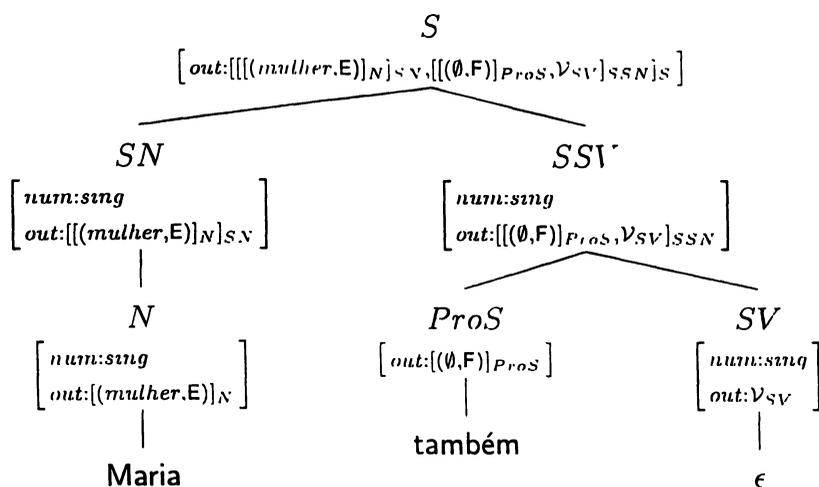
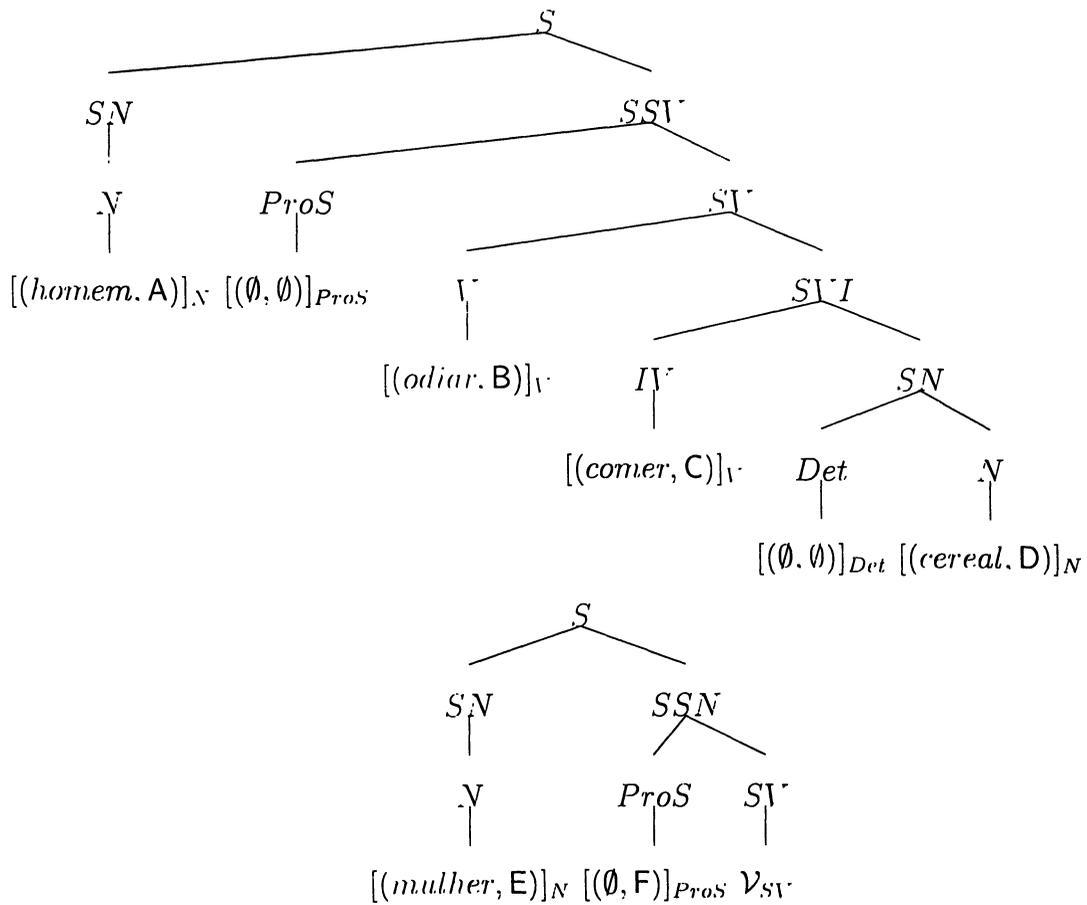


Figura 4.7: Árvore sintática: Maria também.



Executando o algoritmo de complementariedade, obtém-se:

$$C([\emptyset, \emptyset]_{ProS}, [(odiar, B)]_V, [(comer, C)]_V, [(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SV}]_{SSV},$$

$$[(\emptyset, F)]_{ProS}, \mathcal{V}_{SV}]_{SSV}$$

$$[C([\emptyset, \emptyset]_{ProS}, [(\emptyset, F)]_{ProS})]_{SSV} +$$

$$[\bar{C}([(odiar, B)]_V, [(comer, C)]_V, [(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SV}, \mathcal{V}_{SV}]_{SSV} \quad [1]$$

$$[(\emptyset, F)]_{ProS}]_{SSV} +$$

$$[\bar{C}([(odiar, B)]_V, [(comer, C)]_V, [(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SV}, \mathcal{V}_{SV}]_{SSV} \quad [6,11]$$

$$[(\emptyset, F)]_{ProS}]_{SSV} + [taSV]_{SSV} \quad [14]$$

$$[(\emptyset, F)]_{ProS}, [(odiar, B)]_V, [(comer, C)]_V, [(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SV}]_{SSV}$$

Traduzindo a expressão e realizando a conversão beta, obtém-se a representação abaixo.

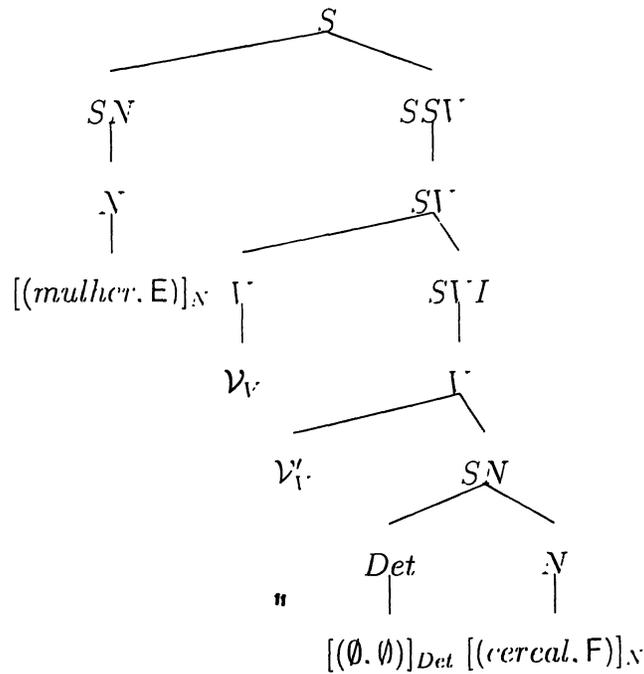
x_1	x_2	e_1	x_3	x_4	e_2
$x_1 = joao$					
$arroz(x_2)$					
$e_1 : comer(x_2)$					
$odeia(x_1, e_1)$					
$x_3 = maria$					
$arroz(x_4)$					
$e_2 : comer(x_4)$					
$odeia(x_3, x_4)$					

Como não há nenhuma ocorrência do operador α , \mathcal{F} simplesmente adiciona os compartimentos de foco e os atualiza, retornando, em seguida, a representação abaixo.

x_1	x_2	e_1	x_3	x_4	e_2
			x_3	x_1	
			x_4	x_2	
$x_1 = joao$					
$arroz(x_2)$					
$e_1 : comer(x_2)$					
$odeia(x_1, e_1)$					
$x_3 = maria$					
$arroz(x_4)$					
$e_2 : comer(x_4)$					
$odeia(x_3, x_4)$					

A solução do exemplo (4.20) é dada a seguir. Assim como o anterior, o algoritmo retorna as árvores sintáticas das duas sentenças. A primeira já foi apresentada na figura 4.6, a segunda é apresentada na figura 4.8.

A representação da segunda sentença do exemplo, retornada pela gramática em forma de árvore, é:



Rodando o algoritmo de complementariedade, obtém-se:

$$\mathcal{C}([[(odiar, B)]_V, [(comer, C)]_V, [[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SSV} \cdot [V]_V \cdot [V']_{V'}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI}]_{SSV}$$

$$[\mathcal{C}([[(odiar, B)]_V, [(comer, C)]_V, [[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}]_{SV}, [V]_V \cdot [V']_{V'}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI}]_{SSV} \quad [1]$$

$$[[\bar{\mathcal{C}}([[(odiar, B)]_V, V]_{SV}) + [\mathcal{C}([[(comer, C)]_V, [[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}, [V']_{V'}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI})]_{SV}]_{SSV} \quad [2]$$

$$[[[(odiar, B)]_V]_{SV} + [\mathcal{C}([[(comer, C)]_V, [[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}]_{SVI}, [V']_{V'}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI})]_{SV}]_{SSV} \quad [14]$$

$$[[[(odiar, B)]_V]_{SV} + [[\bar{\mathcal{C}}([[(comer, C)]_V, V]_{SVI}) + [\mathcal{C}([[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI}]_{SV}]_{SSV} \quad [2]$$

$$[[[(odiar, B)]_V]_{SV} + [[[(comer, C)]_V]_{SVI} + [\mathcal{C}([[(\emptyset, \emptyset)]_{Det}, [(cereal, D)]_N]_{SN}, [[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI})]_{SV}]_{SSV} \quad [14]$$

$$[[[(odiar, B)]_V]_{SV} + [[[(comer, C)]_V]_{SVI} + [[\bar{\mathcal{C}}([[(\emptyset, \emptyset)]_{Det}, [(\emptyset, \emptyset)]_{Det})]_{SN} + [\mathcal{C}([[(cereal, D)]_N, [(cereal, F)]_N]_{SN}]_{SVI}]_{SV}]_{SSV} \quad [2]$$

$$[[[(odiar, B)]_V]_{SV} + [[[(comer, C)]_V]_{SVI} + [[[(\emptyset, \emptyset)]_{Det}]_{\hat{S}\hat{N}} + [\mathcal{C}([[(cereal, D)]_N, [(cereal, F)]_N]_{SN}]_{SVI}]_{SV}]_{SSV} \quad [12]$$

$$[[[(odiar, B)]_V]_{SV} + [[[(comer, C)]_V]_{SVI} + [[[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI}]_{SSV} \quad [3]$$

$$[[[(odiar, B)]_V], [[[(comer, C)]_V], [[[(\emptyset, \emptyset)]_{Det}, [(cereal, F)]_N]_{SN}]_{SVI}]_{SSV}$$

Traduzindo a expressão e realizando a conversão beta, obtém-se a representação abaixo.

x_1	x_2	e_1	x_3	x_4	e_2
$x_1 = joao$					
	$arroz(x_2)$				
		$e_1 : comer(x_2)$			
					$odeia(x_1, e_1)$
$x_3 = maria$					
					$feijao(x_4)$
					$e_2 : comer(x_4)$
					$odeia(x_3, x_4)$

Como não há nenhuma ocorrência do operador α , \mathcal{F} simplesmente adiciona os compartimentos de foco e os atualiza, retornando, em seguida, a representação abaixo.

x_1	x_2	e_1	x_3	x_4	e_2
			x_3	x_1	
			x_4	x_2	
$x_1 = joao$					
					$arroz(x_2)$
					$e_1 : comer(x_2)$
					$odeia(x_1, e_1)$
$x_3 = maria$					
					$feijao(x_4)$
					$e_2 : comer(x_4)$
					$odeia(x_3, x_4)$

O tratamento dos problemas apresentados no capítulo 2 está completo. Para o primeiro deles, o qual nomeei representação do vazio, o tratamento foi parcial. Como a representação dos elementos fóricos em posição de objeto é controlada, apenas os que possuem a característica de serem inanimados passam pelo reconhecedor sintático. A interferência da intransitividade do verbo foi resolvida apenas habilitando o reconhecedor sintático a dar mais de uma interpretação ao verbo. Mais de uma representação é retornada pelo procedimento e assim nenhuma possibilidade é excluída. A identidade entre

objetos e os prossintagmas verbais receberam quase o mesmo tratamento. Como apresentado nesta seção, algumas mudanças na ontologia, na gramática e a adição do algoritmo de complementariedade fizeram este papel.

4.3 Limitações do algoritmo

O capítulo 2 apresenta vários dos problemas a serem enfrentados no tratamento computacional do objeto nulo no Português do Brasil. Alguns deles não são resolvidos pelo algoritmo apresentado aqui.

4.3.1 Regras de aceitação

Cyrino se baseia em pesquisas estatísticas para afirmar que uma zero-anáfora em posição de objeto não pode ocorrer com correferentes animados. De fato, é relativamente difícil encontrar exemplos que admitem o correferente animado. Porém, mesmo sendo raros, eles existem. Abaixo, temos um deles.

(4.21) João trouxe um frango_i. Maria colocou \emptyset _i na geladeira.

(4.22) João trouxe um frango_i. Maria colocou \emptyset _i no galinheiro.

Observe que a substituição de geladeira por galinheiro troca a animosidade do correferente de \emptyset . O exemplo (4.21) traz um objeto inanimado, enquanto que o exemplo (4.22) traz um objeto animado. Aparentemente, não há nada que impeça a aceitação de (4.22) como uma zero-anáfora. Vale lembrar que a zero-anáfora necessariamente identifica os objetos, pois neste caso eles se correferem, enquanto que na elipse de SV não. Observe também que não é possível aceitar que (4.22) seja uma elipse de SV , pois não parece haver complementariedade entre trouxe e colocou.

Agora, suponha que (4.22) pudesse ser tratado como uma elipse de SV . Fazer isso seria o mesmo que afirmar que este discurso possui o mesmo significado que

(4.23) João trouxe [um frango]. Maria colocou [um frango] no galinheiro.

A representação de (4.23), contudo, seria dada pela DRS abaixo.

x_1, x_2, x_3, x_4, x_5
$x_1 = joao$
$frango(x_2)$
$trouxe(x_1, x_2)$
$x_3 = maria$
$frango(x_4)$
$galinheiro(x_5)$
$colocou(x_1, x_4, x_5)$

Aqui, x_2 e x_4 podem não ser o mesmo frango. Ou seja, (4.22) não é uma elipse de SV . Ele deve ser tratado como zero-anáfora.

Outro exemplo que parece quebrar as regras de Cvrino é dado abaixo.

(4.24) Pedro beijou Maria₁. João não beijou \emptyset ₁.

A situação aqui é um tanto interessante. Para tentar evitar que (4.24) seja resolvido como zero-anáfora, pode-se tentar resolvê-lo como uma elipse de SV . Isso parece ser perfeitamente possível, pois os dois SV s são complementares. Seria lhe dado então, um significado idêntico a:

(4.25) Pedro beijou Maria₁. João não beijou Maria₁.

Observe que precisa haver correferência entre os os dois nomes Maria para que o significado permaneça o mesmo. Todavia, da forma como o procedimento de análise semântica está implementado, não é possível esta correferência. Os dois nomes, ainda que idênticos, seriam interpretados como se fossem de duas pessoas diferentes.

Agora, supondo que (4.25) fosse digitado como entrada para o programa, haveria alguma certeza de que as duas Marias são a mesma pessoa? Eu acredito que não. Por esta razão, sou inclinado a pensar que a solução através da elipse de SV não é a mais interessante.

Existem também, exemplos que parecem violar a regra de complementariedade. Abaixo, um deles.

(4.26) Maria comprou uma maçã. Pedro não comeu.

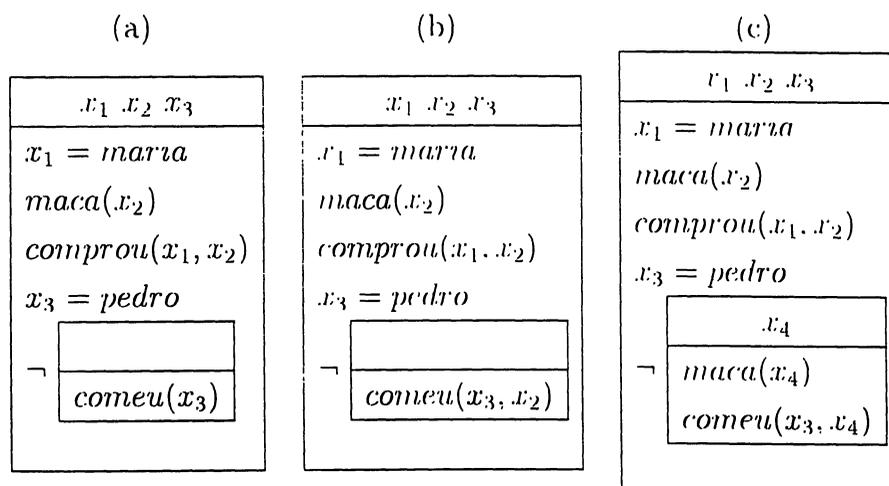
Três interpretações poderiam ser dadas, abaixo, listo todas elas.

(4.27) (a) Maria comprou uma maçã. Pedro não comeu.

(b) Maria comprou uma [maçã]₁. Pedro não [a]₁ comeu.

(c) Maria comprou [uma maçã]. Pedro não comeu [uma maçã].

Observe que os três discursos de (4.27) possuem significados diferentes. Em (4.27a), Maria comprou uma determinada maçã, enquanto que Pedro deixou de comer, no sentido genérico. Em (4.27b), a maçã comprada por Maria é exatamente o mesmo objeto que Pedro deixou de comer. Em (4.27c), Maria comprou uma determinada maçã m_1 , Pedro deixou de comer uma determinada maçã m_2 , mas nada indica que $m_1 = m_2$. Abaixo, as representações em DRT de cada um dos casos.



Enquanto a regra de animosidade permite interpretação (a), não há complementariedade entre os verbos *comprou* e *comeu*. Logo, (b) e (c) não são tratados por esta.

Estes dois casos demonstram a necessidade de estudos ainda mais aprofundados. Outras regras lingüísticas, que não as citadas aqui, parecem permitir o apagamento do objeto nulo no Português do Brasil. É possível que a complementariedade precise ser refinada.

4.3.2 Objetos nulos sentenciais

Sabe-se que alguns verbos aceitam estruturas mais complexas que um *SN* como complemento. Além disso, o Português do Brasil também aceita o apagamento destes objetos, casos estes que são tratados pelo algoritmo proposto anteriormente.

Porém, existem casos de apagamento de constituintes complexos que apresentam problemas muito maiores na sua resolução. Estes casos são chamados de objetos nulos sentenciais.

(4.28) Assim deve ser por força, ainda que não queira \emptyset . (Antonio José, *Guerras do Alecrim e da Manjerona*, p.247)

(4.29) Eu te contei Nilson, que estão procurando uma moça para trabalhar na bomboniere do Pathé?... Eu acho que Margareth podia tentar \emptyset . (Falabella, *No Coração do Brasil* p.9)

Tanto em (4.28) como em (4.29), os discursos carregam informações que não aparecem de forma explícita. Aquilo que Margareth podia tentar, por exemplo, figura como uma referência a ser resolvida. O complemento elidido de (4.29) poderia ser substituído por *trabalhar na bomboniere*, por exemplo. Porém, as teorias baseadas em saliência dos elementos do discurso, restrições, preferências sintáticas e a complementariedade não são capazes de encontrá-lo. Isto se deve ao fato delas não utilizarem um processo de *inferência* capaz de “deduzir”, a partir de informações contidas no discurso ou armazenadas previamente, a resposta para um fenômeno fórico como este. Logo, estes exemplos não podem

ser resolvidos com os procedimentos propostos aqui. Fica evidente que mecanismos mais elaborados precisam ser criados para solucionar casos como este.

4.4 Implementação

Todos os passos do procedimento proposto neste trabalho foram implementados de forma a compor um protótipo capaz de reconhecer discursos em Português. A linguagem utilizada, Prolog, acrescida de uma extensão chamada Gulp ([6]), apresenta um paradigma de programação que traz uma grande vantagem em relação às outras linguagens. Com sua máquina de unificação, tipos de dados dinâmicos e a DCG (*Definite Clause Grammars*), Prolog diminuiu bastante o tempo de desenvolvimento. Além disso, o Gulp facilitou a criação da gramática e dos itens léxicos da linguagem, pois este estende a DCG de Prolog com estruturas de traços para serem utilizadas em gramáticas baseadas em unificação, como é o caso da GPSG. A implementação de todos os procedimentos gerou um código com mais de 3000 linhas, sem contar o código do Gulp.

Abaixo, um exemplo de execução do programa.

```
2 ?- go.
```

```
> maria ama pedro. ele gosta dela.
```

```
drs([a, b], [a, b], [], [object(b), agent(a), gosta(a, b), object(a),
agent(b), ama(b, a), concept(a, homem), gender(a, male), a=pedro,
concept(b, mulher), gender(b, female), b=maria])
```

```
> ana comprou um bolo. maria vendeu.
```

```
drs([a, b, c], [a, c], [b], [agent(a), vendeu(a, b), concept(a, mulher),
gender(a, female), a=maria, object(b), agent(c), comprou(c, b),
concept(b, comida), number(b, singular), gender(b, male), bolo(b),
concept(c, mulher), gender(c, female), c=ana])
```

```
drs([a, b, c, d], [b, d], [a, c], [object(a), agent(b), vendeu(b, a),
concept(a, comida), number(a, singular), gender(a, male), bolo(a),
concept(b, mulher), gender(b, female), b=maria, object(c), agent(d),
comprou(d, c), concept(c, comida), number(c, singular), gender(c, male),
bolo(c), concept(d, mulher), gender(d, female), d=ana])
```

```
> joão odeia comer arroz. maria também.
```

```
drs([a, b, c, d, e, f], [c, f], [], [object(a), agent(c), odeia(c, a),
object(b), concept(a, comer), event(a, comer, b), concept(b, cereal),
number(b, singular), gender(b, male), arroz(b), concept(c, mulher),
gender(c, female), c=maria, object(d), agent(f), odeia(f, d), object(e),
```

```
concept(d, comer), event(d, comer, e), concept(e, cereal),
number(e, singular), gender(e, male), arroz(e), concept(f, homem),
gender(f, male), f=joão]
```

```
> quit.
```

```
Yes
3 ?-
```

A primeira DRS corresponde à seguinte representação:

x_1 x_2
x_2 x_1
1
$x_1 = maria$
$x_2 = pedro$
$ama(x_1, x_2)$
$gosta(x_2, x_1)$

O primeiro detalhe curioso da representação é que ela é dada de traz para frente. Isso ocorre por causa da concatenação das DRSs. Os compartimentos das DRSs, modelados como listas de Prolog, são manipulados como se fossem pilhas. Outro detalhe a ser observado, é que os referentes de discurso que correferem entre si não são igualados, como apresentado nos exemplos até aqui. Na realidade, eles são modelados como variáveis livres da linguagem Prolog. Quando o programa encontra um candidato a correferente que preenche os pré-requisitos, este é unificado com a variável livre que representa o elemento fórico. Desta forma, uma condição como $a = x_1$ nunca é realmente incluída em uma DRS. A variável a é, na verdade, unificada com o referente x_1 . Observe ainda, que as classificações ontológicas dos constituintes estão presentes no segundo compartimento da DRS. Para diferenciá-las das outras, lhes dei o nome de *concept*. Isto é feito desta forma para facilitar o tratamento das correferências fóricas. O algoritmo \mathcal{F} busca as informações ontológicas dos elementos presentes na DRS dentro da própria estrutura.

As duas DRSs apresentadas em seguida correspondem às duas representações possíveis dos exemplos (4.12a) e (4.12b). Uma delas, a primeira, corresponde à representação da zero-anáfora, enquanto que a outra corresponde à representação da elipse de *SV*. Finalmente, a última DRS representa outro exemplo de elipse de *SV* analisado, o exemplo (4.19).

Além das limitações apresentadas na seção 4.3, também existem algumas outras, menos sérias, que se devem à própria maneira com que o código foi implementado. Por exemplo,

sabe-se que a gramática da linguagem é bastante reduzida e, portanto, limitada. Não havia necessidade de incrementá-la demais, pois o protótipo não tinha intenção de ser uma ferramenta de aplicação prática. Logo, algumas ocorrências simples de elipses de *SV* não são aceitas. Para estes casos, o protótipo não retorna resultado algum, pois os discursos não são reconhecidos como instâncias válidas da linguagem. Casos típicos surgem com a inserção de adjuntos, como nos exemplos abaixo.

(4.30) Pedro lava [as mãos com sabão]_i. Maria não lava \emptyset _i.

(4.31) João [come arroz]_j no sofá. Maria \emptyset _j na sala.

Pelas definição de complementariedade, estes são exemplos perfeitamente aceitáveis e tratáveis pelo procedimento. Porém, o protótipo esbarra na limitação do reconhecedor sintático.

Capítulo 5

Conclusão

A linguagem natural, utilizada em larga escala na comunicação entre as pessoas motiva um grande número pesquisas e, no campo estritamente lingüístico, inúmeras teorias sintaticistas tentam explicar os mais diversos fenômenos da linguagem. Com o objeto nulo não é diferente. Independentemente disso, porém, são escassas as teorias semânticas, bem como, as computacionais sobre o assunto. A grande maioria das pesquisas voltadas para a área da computação tratam de outras línguas naturais como Inglês, Francês e Espanhol. Portanto, o presente trabalho volta sua atenção para um fenômeno pouco estudado. Aliado a isto, a relevância do estudo realizado se justifica pela crescente utilização da categoria vazia em posição de objeto no Português no decorrer dos anos, como afirma Cyrino em suas pesquisas estatísticas ([9]). Vale lembrar ainda as já mencionadas aplicações práticas da própria lingüística computacional.

Proponho nesta abordagem um procedimento para construção da representação semântica de discursos que apresentam, não só apagamento do objeto do verbo, como apagamento de alguns outros elementos do sintagma verbal. O encontro dos elementos elididos, a construção de sua representação semântica e a combinação desta com as representações dos demais elementos dos discurso, figuram como passos deste procedimento. Na sua execução, contudo, é possível identificar algumas ambigüidades. Estas ocorrências podem advir de dois fatores. Em primeiro lugar, alguns verbos permitem dupla interpretação quanto sua transitividade. Isso significa que não é possível, ao menos nos níveis sintático e semântico, determinar se o verbo em questão é transitivo ou intransitivo. A outra, pode surgir da dúvida sobre o que exatamente foi apagado no *SV*. Um exemplo contendo estas ambigüidades foi apresentado em (4.27), repetido abaixo.

- (5.1) (a) Maria comprou uma maçã. Pedro não comeu.
(b) Maria comprou uma [maçã]_i. Pedro não [a]_i comeu.
(c) Maria comprou [uma maçã]. Pedro não comeu [uma maçã].

O exemplo (5.1a) interpreta o verbo comeu como intransitivo, enquanto que (5.1b) e (5.1c) interpretam-no como transitivo. Seu complemento, portanto, foi apagado. As

interpretações (5.1b) e (5.1c) diferem no segundo fator. Na primeira delas, assume-se que o complemento apagado é o pronome *a* e na segunda, assume-se o apagamento do *SN* *uma maçã*. Como visto, não se pode escolher entre uma interpretação e outra sem extrapolar o escopo do estudo realizado. Sendo assim, o procedimento assume as três possibilidades, encontrando uma representação semântica diferente para cada uma delas. Para capturar a ambigüidade entre as interpretações transitiva e intransitiva, a gramática possui duas entradas para o verbo *comer*. No caso da outra ambigüidade, os tratamentos diferenciados de zero-anáfora e elipse de *SV* fazem seu papel.

Figuram como contribuições importantes, dadas por este trabalho, a criação de uma gramática capaz de reconhecer uma pequena parte da língua portuguesa, juntamente com vários casos de objeto nulo, sob a forma de zero-anáfora e elipse de *SV*. Uma pequena comparação entre os algoritmos de *centering* ([15]) e de *focusing* ([38]), a qual escolhe *focusing* como preferido. Uma ligeira modificação e acréscimos à ontologia proposta por Saiz-Noeda e Palomar ([35]) afim de utilizar o conhecimento de mundo na resolução das elipses de *SV*. A criação de um algoritmo genérico para resolução de fenômenos fóricos que implementa as verificações de gênero, número, consistência semântica e saliência utilizando *focusing*. A implementação das técnicas propostas por Cyrino ([9]) para diferenciação entre zero-anáforas em posição de objeto e elipses de *SV*. Uma implementação para tratamento de elipses de *SV* no nível sintático através da proposição do conceito de complementariedade, utilizando neste, informações sintáticas e de conhecimento de mundo. A fusão de todas estas técnicas com a proposta composicional de construção de representações semânticas em DRT dada por Muskens ([29]).

Este trabalho também aponta vários problemas ainda não resolvidos. As regras de aceitação do objeto nulo, ainda que apoiadas por pesquisas de corpus, se mostram incompletas diante de alguns exemplos apresentados na seção 4.3.1. Os problemas apresentados pelo contexto lingüístico e por referências sentenciais demonstram a importância do tratamento semântico a ser dado para elementos fóricos. Contar com alguma informação semântica, dada pelo conhecimento de mundo ou pela representação do discurso, é de grande valia para sua interpretação. Além disso, sua solução exige conhecimento pragmático e técnicas de inferência.

Como possíveis trabalhos futuros, pode-se sugerir, justamente, o refinamento das regras de animosidade e complementariedade, com o objetivo de solucionar casos de apagamento ainda sem tratamento adequado. Um desafio ainda mais interessante seria a incorporação de informações do nível pragmático e a aplicação de técnicas de inferência para solução dos objetos nulos sentenciais. Vale lembrar ainda, que a implementação construída não serve como uma aplicação prática dos conceitos propostos. É possível estendê-la incluindo novas regras gramaticais e acrescentando novos itens ao léxico.

Referências Bibliográficas

- [1] José Abraços e José Gabriel Lopes. Extending DRT with a focusing mechanism for pronominal anaphora and ellipsis resolution, 1994.
- [2] James Allen. *Natural Language Understanding*. Benjamin/Cummings, Redwood. 2 edition, 1995.
- [3] Patrick Blackburn e Johan Bos. Representation and inference in natural language: A first course in computational semantics. volume II: Working with discourse representations. Available at <http://www.comsem.org>. 1999.
- [4] Susan E. Brennan, Walker Friedmanand, e Carl J. Pollard. A centering approach to pronouns. *25th Annual Meeting of the Association of Computational Linguistics*, páginas 155–162, 1987.
- [5] Alonzo Church. *The calculi of lambda-conversion*. Princeton University, Princeton – USA, 1941.
- [6] Michael A. Covington. Gulp 3.1: An extension of prolog for unification based grammar. Relatório técnico. Artificial Intelligence Center, The University of Georgia. Athens, Georgia U.S.A.. 1994. Research report AI 1994 06.
- [7] Sonia Maria Lazzarini Cyrino. *O objeto nulo no Português do Brasil: uma investigação diacrônica*. UNICAMP, 1990.
- [8] Sonia Maria Lazzarini Cyrino. *O objeto nulo no Português do Brasil: uma mudança paramétrica?* UNICAMP, 1990.
- [9] Sonia Maria Lazzarini Cyrino. *O objeto nulo no Português do Brasil*. UEL, Londrina, 1997.
- [10] David R. Dowty, Robert E. Wall, e Stanley Peters. *Introduction to Montague Semantics*, volume 1 of *Synthese language library*. D. Reidel Publishing Co., Dordrecht, 1981.
- [11] M. E. L. Duarte. *Variação e sintaxe: Clítico acusativo, pronome lexical e categoria vazia do Português do Brasil*. Dissertação de Mestrado. PUC-SP, 1986.
- [12] Gerald Gazdar, Klein Ewan. Geoffrey K. Pullum. e Ivan Sag. *Generalized Phrase Structure Grammar*. Atlas. Oxford, 1985.
- [13] Peter T. Geach. *Reference and Generality : An Examination of Some Medieval and Modern Theories*. Cornell University Press. 1962 (Third revised edition: 1980).

- [14] Algirdas Julien Greimas e Joseph Courtés. *Dicionário de Semiótica*. São Paulo: Cultrix, 1989.
- [15] Barbara J. Grosz, Aravind K. Joshi, e Scott Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, volume 21, páginas 203–225. junho de 1995.
- [16] Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. Tese de Doutorado, Linguistics Department, University of Massachusetts, Amherst, Massachusetts, 1982.
- [17] Rodolfo Ilari. Os reflexivos e a interpretação dos prossintagmas. *Cadernos de Estudos Lingüísticos*. (2):56–156. 1981.
- [18] Hans Kamp. A theory of truth and semantic interpretation. Jeroen A. G. Groenendijk, T. M. V. Janssen, e Martin B. J. Stokhof, editors. *Formal methods in the study of language*, páginas 277–322. Mathematical Centre Tracts. Amsterdam, 1981.
- [19] Hans Kamp e Uwe Reyle. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht, 1993.
- [20] Dimitar Kazakov, Stephen Pulman, e Stephen Muggleton. The FraCaS dataset and the LLL challenge. ILP2 project paper, julho de 1998.
- [21] Richard Larson e Gabriel Segal. *Knowledge of Meaning: An Introduction to Semantic Theory*. The MIT Press, Cambridge, Massachusetts, 1995.
- [22] Tiago Santos de Lima. Abordagem composicional para interpretação semântica de discurso em língua portuguesa. Trabalho de graduação, 2001.
- [23] Tiago Santos de Lima e Michel Gagnon. Uma implementação da teoria de *Centering* para resolução de anáforas em língua portuguesa. Relatório Técnico RT_DINF 003/2002, Universidade Federal do Paraná, Curitiba, 2002.
- [24] Gabriela Matos e Sónia Cyrino. Eclipse de VP no Português Europeu e no Português brasileiro. *Anais do 2º Congresso Internacional da Abralín*. março de 2001.
- [25] M. G. A. P. Matos. *Construções de Eclipse de Predicado em Português – SV Nulo e Despojamento*. Tese de Doutorado, Universidade Federal de Lisboa, 1992.
- [26] Ruslan Mitkov. Anaphora resolution: The state of the art. Artigo baseado no tutorial em resolução de anáforas COLING'98/ACL'98, 1999.
- [27] Richard Montague. The proper treatment of quantification in ordinary english. *Approaches to Natural Language*. D. Reidel Publishing Co., Dordrecht, Holland, 1973.
- [28] Richard Montague. English as formal language. *Selected Papers of Richard Montague*, páginas 108–221. Yale University Press, 1974.
- [29] Reinhard Muskens. Combining Montague semantics and discourse representation. *Linguistics and Philosophy*. 19(2):143–186. 1996.
- [30] Nelize Pires de Omena. Pronome pessoal de terceira pessoa: suas formas variantes em função acusativa. Dissertação de Mestrado, PUC-RJ, Rio de Janeiro, 1978.

- [31] Ivandré Paraboni. Uma arquitetura para a resolução de referências pronominais possessivas no processamento de textos em língua portuguesa. Dissertação de Mestrado. Pontifícia Universidade Católica do Rio Grande do Sul. Porto Alegre, dezembro de 1997.
- [32] Ivandré Paraboni e Vera Lúcia Strube de Lima. Possessive pronominal anaphor resolution in portuguese written texts. *Proceedings of the 17th International Conference on Computational Linguistics*. páginas 1010–1014. Montreal, agosto de 1998.
- [33] Tatiana de Oliveira Petry. Uso da teoria de centering no tratamento computacional de referências anafóricas pronominais no Português escrito: Uma experimentação com pronomes pessoais. Dissertação de Mestrado, PUC-RS. Porto Alegre, 2000.
- [34] Eduardo Raposo. On the null object in european portuguese. *Studies in Romance Linguistics*. C. Neidle & R. Cedeño. Dordrecht, 1986.
- [35] Maximiliano Saiz-Noeda e Manuel Palomar. Semantic knowledge-driven method to solve pronominal anaphora in Spanish texts. *Lecture Notes in Computer Science*. 1835:204–211, 2000.
- [36] Candace L. Sidner. Focusing for interpretation of pronouns. *American Journal of Computational Linguistics*. 7(4):217–231, 1981.
- [37] Candace L. Sidner. Focusing in the comprehension of definite anaphora. Robert C. Berwick e Michael Brady, editors. *Computational Models of Discourse*. capítulo 5, páginas 267–330. MIT Press. Cambridge, MA, 1983. Q335 .C56 1983.
- [38] Candance L. Sidner. Towards a computational theory of definite anaphora comprehension in English discourse. Relatório Técnico AI-TR-537, MIT, 1979.
- [39] The FRACAS Consortium. Describing the approaches, dezembro de 1994.
- [40] Jan van Eijck e Hans Kamp. Representing discourse in context. *Handbook of Logic and Linguistic*. Elsevier, 1996.
- [41] George Yule. *The study of Language*. Cambridge University Press, Cambridge, 1996.

Índice Remissivo

- + , *veja* operador de concatenação
- \in , *veja* compatibilidade
- \in_{ad} , *veja* compatibilidade entre sujeito e predicativo
- \in_{ob} , *veja* compatibilidade entre verbo e complemento verbal
- $\in_{s\gamma}$, *veja* compatibilidade entre verbo e sujeito
- \oplus , *veja* operador fusão
- $*$, \bar{n}
- \emptyset , *veja* objeto nulo
- O_a , 56
- O_n , 34
- O_v , 35
- R_{ad} , 58
- R_{ob} , 35
- $R_{s\gamma}$, 35
- α , *veja* operador alfa
- ϵ , *veja* palavra vazia
- λ , *veja* operador lambda

- α -DRT. 41
- λ -DRS. 28

- A, 64. 68
- abreviações de DRSs, 21
- abstração lambda. 27
- acessibilidade, 24
- adjetivo, 68
- algoritmo
 - de *focusing*, 38
 - de atualização dos focos atores. 39
 - de atualização dos focos de discurso, 39
 - de complementariedade – C . 67
 - de escolha dos referentes, 38
 - de representação de discursos. 73
 - de resolução de fenômenos fóricos – \mathcal{F} , 48
 - de solução de elipses de SV – \mathcal{R} . 67
 - de tradução – \mathcal{T} , 66

- âncora. 40
- antecedente, 7
- aplicação (expressão lambda). 27
- atribuição. 22

- C, 15. 30. 64
- cálculo lambda. 27
- centering*, 39
- centro. 39
- centro prospectivo. 39
- centro retrospectivo, 39
- compatibilidade, 35
 - entre adjetivos, 58
 - entre condições, 45
 - entre substantivos, 56
 - entre sujeito e predicativo. 58
 - entre verbo e complemento verbal. 35
 - entre verbo e sujeito. 35
 - entre verbos, 56
- condição, 20
- condição de verdade de DRS, 23
- conectivo, *veja* C
- conhecimento de mundo. 34. 55
- consistência semântica, 33
- continuing*, 40
- contra-índice. 33
- convenções lambda, 28
- conversão beta, 27
- correferência, 6

- D, 14, 15, 29, 62
- Det*, 63, 68
- determinante, *veja* *Det*
- discurso, 13, 14, *veja* D
- domínio. 22
- donkey sentence*, 18
- DRS. 20
- DRT. 19

- elemento fórico. 6
- elipse de *SV*, 52, 61

- escopo. 27
- estrutura de representação de discurso.
 veja DRS
- estrutura de traços. 14
- expressão lambda. 27
- extensão (atribuição). 22
- fenômeno fórico. 2. 6. 32
- foco. 37
- foco ator. 37
- foco de discurso. 37
- foco-DRS. 46
- focusing*. 37
- função de interpretação. 22
- funtor. 27
- Generalized Phrase Structure Grammar* –¹
 GPSG. 14
- Gramática 1, 14
- Gramática 2, 29
 adendos à. 42
 modificação da. 58
- Gramática 3, 62
- Mod.* 65. 67
- modelo. 22
- modificador. *veja Mod*
- N*, 15, 29. 30, 42, 64
- objeto nulo, 2, 6, 7, 50
- ontologia. *veja* conhecimento de mundo
- operador alfa, 41
- operador de concatenação, 68
- operador fusão, 25. 29
- operador lambda, 27
- palavra vazia, 8
- paralelismo, 35
- PB, 3
- PE, 3
- próximo centro preferencial, 39
- preferência, 32
- prefixo. 27
- princípio da composicionalidade. 13
- Pro*, 42
- problema da ligação anafórica do contexto.
 18
- pronome. *veja Pro*
ProS, 64. 67
- prossintagma verbal. 11. *veja ProS*
- referente de discurso, 20
- regra de apagamento de *SV*. 54
- restrição. 32
- retaining*, 40
- rough-shifting*, 40
- S*, 14. 15. 29. 42. 62
- saliência, 36
- satisfação de condição, 22
- satisfação de DRS. 22
- sentença, 14, *veja S*
shifting, 40
- sintagma. 14
- sintagma nominal. *veja SN*
- sintagma verbal. *veja SV*
- sintagma verbal infinitivo, *veja SVI*
SN, 15, 29, 58, 62, 63
SSV, 63, 67
- subordinação de DRSs, 24
- substantivo, *veja N*
- super sintagma verbal, *veja SSV*
SV, 15, 29, 63
SVI, 63
- teoria de representação de discurso, *veja*
 DRT
- termo, 20
- transição vazia, 7
- V*, 15. 30. 64
- variável (expressão lambda), 27
- variável ligada. 27
- verbo, *veja V*
- Z*, 64
- zero-anáfora. 2, *veja Z*, 52, 58