

GABRIEL DE ÁVILA OTHERO

# A GRAMÁTICA DA FRASE EM PORTUGUÊS

ALGUMAS REFLEXÕES PARA A FORMALIZAÇÃO  
DA ESTRUTURA FRASAL EM PORTUGUÊS

***A GRAMÁTICA DA FRASE EM PORTUGUÊS***  
***ALGUMAS REFLEXÕES PARA A FORMALIZAÇÃO DA ESTRUTURA FRASAL***  
***EM PORTUGUÊS***



**Pontifícia Universidade Católica do Rio Grande do Sul**

**Chanceler:**

*Dom Dadeus Grings*

**Reitor:**

*Joaquim Clotet*

**Vice-Reitor:**

*Evilázio Teixeira*

**Conselho Editorial:**

*Antônio Carlos Hohlfeldt*  
*Elaine Turk Faria*  
*Gilberto Keller de Andrade*  
*Helenita Rosa Franco*  
*Jaderson Costa da Costa*  
*Jane Rita Caetano da Silveira*  
*Jerônimo Carlos Santos Braga*  
*Jorge Campos da Costa*  
*Jorge Luis Nicolas Audy (Presidente)*  
*José Antônio Poli de Figueiredo*  
*Jussara Maria Rosa Mendes*  
*Lauro Kopper Filho*  
*Maria Eunice Moreira*  
*Maria Lúcia Tiellet Nunes*  
*Marília Costa Morosini*  
*Ney Laert Vilar Calazans*  
*René Ernaini Gertz*  
*Ricardo Timm de Souza*  
*Ruth Maria Chittó Gauer*

**EDIPUCRS:**

*Jerônimo Carlos Santos Braga – Diretor*  
*Jorge Campos da Costa – Editor-chefe*

*Gabriel de Ávila Othero*

***A GRAMÁTICA DA FRASE EM PORTUGUÊS***  
***ALGUMAS REFLEXÕES PARA A FORMALIZAÇÃO DA ESTRUTURA FRASAL***  
***EM PORTUGUÊS***



PORTO ALEGRE  
2009

© EDIPUCRS, 2009

Capa: Vinícius de Almeida Xavier

Diagramação: Gabriela Viale Pereira

Revisão Linguística: Grasielly Hanke Angeli

### **Dados Internacionais de Catalogação na Publicação (CIP)**

O87g Othero, Gabriel de Ávila

A gramática da frase em português [recurso eletrônico] :  
algumas reflexões para a formalização da estrutura frasal em  
português / Gabriel de Ávila Othero. – Dados eletrônicos. – Porto  
Alegre : EDIPUCRS, 2009.

160 p.

Sistema requerido: Adobe Acrobat Reader

Modo de Acesso: World Wide Web:

<<http://www.pucrs.br/orgaos/edipucrs/>>

ISBN 978-85-7430-854-8 (on-line)

1. Português – Gramática. 2. Português – Análise Sintática. 3.  
Linguística Aplicada. 4. Linguística Computacional. I. Título.

CDD 410.285

**Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS**



Av. Ipiranga, 6681 - Prédio 33  
Caixa Postal 1429  
90619-900 Porto Alegre, RS - BRASIL  
Fone/Fax: (51) 3320-3711  
E-mail: [edipucrs@pucrs.br](mailto:edipucrs@pucrs.br)  
<http://www.pucrs.br/edipucrs>

## **SOBRE O AUTOR**

**Gabriel de Ávila Othero** é Doutor em Linguística Aplicada pela Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS. O autor tem quatro livros publicados, além de artigos acadêmicos em periódicos nacionais. Também é o editor-fundador da Revista Virtual de Estudos da Linguagem – ReVEL.

## **AGRADECIMENTOS**

Este livro eletrônico passou pela leitura crítica de diversos colegas e professores. Alguns leram o texto integralmente; outros leram alguns capítulos. Todos, porém, colaboraram para minha redação final do livro. Deixo aqui registrado meu agradecimento às seguintes pessoas: Ana T. Ibaños, Carlos Prolo, João Paulo Cyrino, Jorge Campos Costa, Leda Bisol, Magdiel M. Aragão Neto, Mário A. Perini, Mathias Schaff Filho e Sérgio de Moura Menuzzi.

Os erros e inconsistências que persistem são de minha inteira responsabilidade.

## LISTA DE FIGURAS, QUADROS E TABELAS

<b>Figura 3.1:</b> Exemplo de sentença analisada com colchetes rotulados .....	36
<b>Figura 3.2:</b> Exemplo de sentença analisada com sua estrutura arbórea.....	36
<b>Quadro 3.1:</b> Verbos no léxico do Grammar Play .....	42
<b>Quadro 3.2:</b> Substantivos e adjetivos no léxico do Grammar Play .....	42
<b>Quadro 3.3:</b> Regras da sentença .....	43
<b>Quadro 3.4:</b> Regras do NP.....	43
<b>Quadro 3.5:</b> Regras do AP.....	44
<b>Quadro 3.6:</b> Regras do PP .....	44
<b>Quadro 3.7:</b> Regras do VP.....	45
<b>Quadro 3.8:</b> Regras do AdvP .....	46
<b>Quadro 4.1:</b> Regras de Lemle (1984) e Lobato (1986) .....	113
<b>Quadro 4.2:</b> Regras gramaticais dos agrupamentos nominais .....	142
<b>Quadro 4.3:</b> Regras gramaticais dos sintagmas adjetivais.....	142
<b>Quadro 4.4:</b> Regras gramaticais dos sintagmas preposicionais .....	142
<b>Quadro 4.5:</b> Regras gramaticais dos sintagmas adverbiais.....	142
<b>Quadro 4.6:</b> Regras gramaticais dos sintagmas verbais .....	143
<b>Quadro 4.7:</b> Regras gramaticais da sentença .....	143
<b>Tabela 2.1:</b> Núcleos lexicais .....	30

## LISTA DE SÍMBOLOS E ABREVIATURAS

- ∅ = elemento vazio
- Adj = adjetivo
- Adv = advérbio
- AdvP = sintagma adverbial
- AP = sintagma adjetival
- CFG = *Context-free grammar*
- CP = *complementizer phrase*
- DCG = *Definite Clause Grammars*
- Det = determinante
- DP = *determiner phrase*
- GB = Teoria da Regência e Ligação
- HPSG = *Head-driven Phrase Structure Grammar*
- IP = *inflectional phrase*
- LFG = *Lexical Functional Grammar*
- N = nome (ou substantivo)
- NP = sintagma nominal
- NumP = sintagma numeral
- P = preposição
- PB = português brasileiro
- PLN = Processamento de Linguagem Natural
- PossP = sintagma possessivo
- PP = sintagma preposicional
- PSG = *Phrase Structure Grammar*
- QP = *quantifier phrase*
- S = sentença
- V = verbo
- VP = sintagma verbal

*Para quem gosta de certezas e seguranças, tenho más notícias: a gramática não está pronta. Para quem gosta de desafios, tenho boas notícias: a gramática não está pronta. Um mundo de questões e problemas continua sem solução, à espera de novas idéias, novas teorias, novas análises, novas cabeças.*

Mário A. Perini

## SUMÁRIO

	<b>PRIMEIRÍSSIMAS PALAVRAS.....</b>	<b>11</b>
<b>1</b>	<b>UMAS PRIMEIRAS PALAVRAS .....</b>	<b>12</b>
<b>2</b>	<b>O FORMALISMO GRAMATICAL.....</b>	<b>20</b>
2.1	A GRAMÁTICA SINTAGMÁTICA .....	20
2.2	TEORIA X-BARRA .....	24
2.3	AS PROJEÇÕES MÁXIMAS EM NOSSAS ANÁLISES.....	30
<b>3</b>	<b>O GRAMMAR PLAY E AS ANÁLISES DE OTHERO (2004).....</b>	<b>35</b>
<b>4</b>	<b>ANÁLISES DOS SINTAGMAS DO PB .....</b>	<b>47</b>
4.1	O SINTAGMA NOMINAL.....	47
<b>4.1.1</b>	<b>Sintagmas nominais com modificadores e complementos oracionais .....</b>	<b>82</b>
<b>4.1.2</b>	<b>Alguns problemas na descrição dos sintagmas nominais .....</b>	<b>86</b>
4.1.2.1	Advérbio como modificador nominal .....	87
4.1.2.2	Elementos pesados no sintagma nominal.....	91
4.2	O SINTAGMA ADJETIVAL.....	92
4.3	O SINTAGMA PREPOSICIONAL.....	102
4.4	O SINTAGMA ADVERBIAL.....	109
4.5	O SINTAGMA VERBAL.....	112
4.6	ANÁLISE DA SENTENÇA EM PB .....	124
<b>4.6.1</b>	<b>Sentenças com verbos auxiliares.....</b>	<b>125</b>
<b>4.6.2</b>	<b>Sentenças simples .....</b>	<b>131</b>
<b>4.6.3</b>	<b>Sentenças com dois VPs.....</b>	<b>135</b>
<b>4.6.4</b>	<b>IPs e CPs como complementos do verbo.....</b>	<b>136</b>
<b>4.6.5</b>	<b>Passivas .....</b>	<b>140</b>
4.7	RESUMO DA GRAMÁTICA .....	141
	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>144</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>146</b>

## **PRIMEIRÍSSIMAS PALAVRAS**

Uma professora de Semântica disse uma vez algo mais ou menos assim à nossa turma: “estudar ciência e fazer Linguística é, antes de tudo, revelar problemas. Vocês devem se dar por satisfeitos se saírem de nosso curso com mais dúvidas do que tinham quando chegaram. Aqueles de vocês que acham que vieram aqui apenas para buscar respostas ficarão decepcionados. Vocês verão que sairão do curso com mais dúvidas e questionamentos do que quando chegaram. E aí então, eu ficarei satisfeita”.

Este livro tem também esse caráter. O que as próximas páginas irão mostrar é o resultado de um trabalho cuidadoso, modesto, honesto e atento que reflete minhas dúvidas enquanto estudioso da gramática da língua portuguesa. Meu trabalho irá trazer alguns questionamentos meus que tentei solucionar, alguns problemas já conhecidos por gramáticos e linguistas e alguns desafios que os linguistas computacionais por vezes enfrentam ao trabalhar com a sintaxe das línguas naturais e, especialmente, da língua portuguesa. Muitos dos problemas ficarão sem solução, seja porque eu não tive a capacidade de resolvê-los, seja porque permaneci fiel demais à teoria que adotei para trabalhar com os dados, mesmo quando a teoria se mostrava limitada frente aos fatos da língua.

De qualquer forma, espero que este trabalho possa servir também para guiar novas dúvidas e novas discussões sobre novos e velhos desafios no estudo da sintaxe da língua portuguesa. Sigo Mário Perini quando afirma que “para um linguista não é vergonha dizer ‘não sei’” (Perini, 2006: 186) e assim dou por iniciado este livro.

## 1 UMAS PRIMEIRAS PALAVRAS

*Rem tene, verba sequentur*<sup>1</sup>.

Marcus Porcius Cato

Ray Jackendoff conta uma piada já consagrada entre os linguistas computacionais: sempre que os informatas resolvem pedir auxílio aos linguistas teóricos em algum de seus projetos de Processamento de Linguagem Natural (PLN), seus programas acabam se tornando menos eficientes<sup>2</sup>. Isso porque, como veremos, esses dois tipos de pesquisadores nem sempre têm os mesmos objetivos no estudo da linguagem humana. Aristóteles já havia afirmado que “o objetivo do conhecimento teórico é a verdade, enquanto o do conhecimento prático é a eficácia”<sup>3</sup>. Utilizando-nos de uma comparação, podemos nos arriscar a dizer que o linguista tem um compromisso com a verdade, e o informatas, com a eficácia<sup>4</sup>.

As análises que iremos desenvolver aqui tentarão olhar para os dois lados da Avenida Linguística Computacional. Ou seja, por um lado, iremos nos preocupar em descrever a estrutura interna da sentença em português brasileiro com base em um formalismo linguístico consagrado nos estudos sintáticos (a saber, o formalismo da *Phrase Structure Grammar*, nos moldes propostos pela teoria X-barras). Por outro lado, estaremos também pensando em uma possível e potencial aplicação de nossas análises em aplicativos que envolvam o processamento sintático computacional. Nosso estudo certamente será uma matéria da Sintaxe, mas estaremos sempre “de olho” em uma possível implementação computacional. Iremos propor uma descrição sintática da sentença em português (na verdade, de alguns tipos de estruturas sentenciais) com a pretensão de auxiliar não apenas o linguista, mas também o informatas e o linguista computacional que estiverem interessados na compreensão da estrutura sintática interna da sentença e dos sintagmas em PB. Partiremos de análises já consagradas na literatura sobre a sintaxe do português (especialmente Pontes, 1973; Lobato, 1975, 1986; Perini, 1989, 2000; Lemle, 1984; Luft, 1986; Raposo, 1992; e Miotto et al., 2004), além de análises formais voltadas à aplicação

---

<sup>1</sup> “Domine o assunto, e as palavras irão fluir”.

<sup>2</sup> Jackendoff (2002: 267): “Computational linguists, I am told, joke that every time they hire a theoretical linguist, their programs become less effective”.

<sup>3</sup> Aristóteles (1969 [350 a.C.]), livro II, parte 1.

<sup>4</sup> Perseguimos essa ideia com algum aprofundamento em Othero (2008).

computacional, como Othero (2004 e 2006), Pagani (2004) e David (2007), e apresentaremos nossas análises, que tentarão ser mais precisas, abrangentes e coerentes do que as análises que estudamos.

Como dissemos, argumentaremos que nossas regras gramaticais poderão ser interessantes tanto ao linguista computacional quanto ao sintaticista formal. Isso porque elas serão elaboradas, por um lado, pensando na implementação computacional (mostraremos um exemplo de implementação computacional de regras sintagmáticas do português no terceiro capítulo) e, por outro lado, elas serão sintaticamente consistentes e seguirão um modelo largamente utilizado em teoria gramatical, o formalismo da teoria X-barra.

\*\*\*

Há uma pergunta que todo o pesquisador deve saber responder sobre seu próprio trabalho (e tentaremos respondê-la ao longo deste livro, mas especialmente aqui neste capítulo): qual é a relevância do seu trabalho? Ou seja, qual é a sua importância?

Eis aí uma boa pergunta, porém ambígua: a pergunta pode se referir a nosso livro, especificamente, ou à área em que o livro se insere. Em outras palavras, a pergunta pode ser entendida como “por que nosso texto é relevante? Qual é a sua contribuição original?” ou como “por que a descrição e o processamento sintático das línguas naturais – e do português, em específico – são relevantes?”.

Já que temos tempo e espaço não nos falta, nos arriscaremos<sup>5</sup> a responder às duas perguntas. Debrucemo-nos primeiramente sobre a última: qual é a importância de se estudar a língua de maneira formal, matematizada (por que não?), pensando em sua implementação computacional? E mais especificamente: qual é a importância de se estudar sobre a descrição e o processamento sintático de uma língua?

O processamento sintático é uma área de interface entre a Sintaxe e a Linguística Computacional. Esta última é a área da ciência Linguística preocupada com o tratamento computacional da linguagem e das línguas naturais. Ela envolve diversos campos de estudo da Linguística Teórica e Aplicada, e seus primeiros desenvolvimentos datam dos anos 1950. A área teve grande impulso principalmente graças a esforços para o desenvolvimento de programas de tradução automática nas décadas de 1950 e 1960 e está

---

<sup>5</sup> Decidimos conscientemente evitar o uso da mesóclise em nosso texto.

intrinsecamente ligada a desenvolvimentos na área da Inteligência Artificial<sup>6</sup>.

De acordo com Grisham (1992: 1),

o potencial para o processamento de linguagem natural foi reconhecido bem cedo no desenvolvimento de computadores, e trabalhos em Linguística Computacional – basicamente para tradução automática – começaram na década de 1950 em diversos centros de pesquisa. O rápido crescimento na área, no entanto, aconteceu principalmente a partir do final dos anos 1970.<sup>7</sup>

E a Sintaxe tem se mostrado uma área da Linguística de extremo valor para os desenvolvimentos em Linguística Computacional. O nível de formalização sintática alcançado pelas pesquisas em Linguística tem proporcionado bons resultados não somente para pesquisas de cunho teórico envolvendo o estudo de línguas naturais, mas também para o desenvolvimento de aplicativos computacionais, como *parsers*, *taggers*, corretores ortográficos e gramaticais, tradutores automáticos, etc.<sup>8</sup> Para Sag, Wasow & Bender (2003: 14), “uma das áreas mais promissoras para a aplicação da teoria sintática é no desenvolvimento de tecnologias robustas que lidam com a linguagem natural”<sup>9</sup>.

O estudo da sintaxe de uma língua pode ser essencial para seu tratamento computacional em diversos outros níveis de descrição linguística, e normalmente trabalhos de implementação computacional em uma língua envolvem o tratamento sintático ou morfossintático dessa língua. Para Sag, Wasow & Bender (2003: 14),

até mesmo quando tecnologias usáveis que lidem com a linguagem natural podem ser desenvolvidas sem se utilizarem da pesquisa em gramática, normalmente o que acontece é que elas podem se tornar mais robustas ao incluírem um componente de base sintática.<sup>10</sup>

---

<sup>6</sup> Cf. Sparck Jones (1994), Russel & Norvig (1995) e Hutchins (2000).

<sup>7</sup> Trecho original: “The potential for natural language processing was recognized quite early in the development of computers, and work in computational linguistics – primarily for machine translation – began in the 1950s at a number of research centers. The rapid growth in the field, however, has taken place mostly since the late 1970s”.

<sup>8</sup> Cf. Nijholt & Sikkil (1993), Bick (1996) e Garside, Leech and McEnery (1997).

<sup>9</sup> Trecho original: “one of the most promising areas for applying syntactic research is in the development of useful and robust natural language technologies”.

<sup>10</sup> Trecho original: “even where usable natural language technologies can be developed that are not informed by grammatical research, it is often the case that they can be made more robust by including a principled syntactic component”. Porém, o processamento sintático não é essencial. Wong & Wu (1999), por exemplo, defendem a ideia de que o desenvolvimento de *parsers* robustos não precisa sequer contar com gramáticas e regras sintáticas: “Construir uma gramática que defina/descreva precisamente uma língua natural é muito difícil, se não impossível. Uma gramática construída à mão ou induzida automaticamente costuma ser incompleta e generalizadora demais. (...) Uma gramática explícita [para efetuar o *parsing* de uma língua] não é necessária. (...) [Acreditamos na] possibilidade de melhorar a precisão e a velocidade de *parsing* se eliminarmos uma gramática explícita” (p. 121-2). Ver também, a esse respeito, Kac & Manaster-Ramer (1986), que admitem que “em Linguística Computacional, existe uma forte (se não universal) dependência em gramáticas sintagmáticas como meio de se representar a estrutura sintática de uma língua” (p. 156); eles

O fato de se estudar a sintaxe de uma língua antes de partir para outros tipos de estudos em sua formalização também pode estar relacionado à **sintaticização**, como chama Culicover (1997). Ou seja, costumamos pensar que o tratamento sintático pode preceder outros estudos formais da linguagem. Para Culicover (1997: 12), a sintaticização é

a tática de pegar qualquer fenômeno que esteja plausivelmente ligado à estrutura sintática e caracterizá-lo em termos sintáticos. A sintaticização é, no mínimo, uma maneira de tornar materiais lingüísticos complexos passíveis de análise e de produzir hipóteses e previsões claras.<sup>11</sup>

De qualquer forma, concordamos com o que dizem Sag, Wasow & Bender (2003: 16):

Desenvolver tecnologias realmente robustas que lidem com a linguagem natural – ou seja, programas que irão te permitir interagir com o computador na TUA língua e não na língua do computador – requer análises detalhadas e cuidadosas sobre a estrutura gramatical da língua e como ela influencia o significado. Atalhos que se baseiem em heurísticas, tentativas de adivinhação ou simples usos de *templates* irão inevitavelmente levar a erros.<sup>12</sup>

Para Bateman, Forrest & Willis (1997: 166),

um dos principais objetivos da área de PLN nos últimos dez anos tem sido produzir um “analisador gramatical”, ou *parser*, de **abrangência ampla**. Para muitos aplicativos de PLN, o desafio é produzir um *parser* que poderá ser capaz de analisar automática e estruturalmente de maneira correta, de acordo com um esquema de *parsing* definido, qualquer sentença do inglês que possa ocorrer naturalmente, **sem restrições**, de uma gama de gêneros textuais tão vasta quanto possível.<sup>13</sup> (grifos dos autores).

Doze anos já se passaram desde que essas linhas de Bateman, Forrest & Willis

---

pretendem mostrar que as gramáticas sintagmáticas e as regras gramaticais de reescrita são desnecessárias e redundantes para as representações lingüísticas de que um *parser* necessita.

<sup>11</sup> Trecho original: “Syntacticization is the tactic of taking every phenomenon that is plausibly tied to syntactic structure and characterizing it in syntactic terms. At the very least, Syntacticization is a way of rendering complex linguistic material amenable to analysis and of producing clear hypotheses and predictions.”

<sup>12</sup> Trecho original: “Building truly robust natural language technologies – that is, software that will allow you to interact with your computer in YOUR language, rather than in ITS language – requires careful and detailed analysis of grammatical structure and how it influences meaning. Shortcuts that rely on semantic heuristics, guesses, or simple pattern-matching will inevitably make mistakes”.

<sup>13</sup> Trecho original: “one of the major aims of NLP over the past ten years has been to produce a **wide-range** ‘grammatical analyser’ or **parser**. For many NLP applications, the challenge is to produce a parser which will automatically be able to structurally analyse correctly, according to a defined *parsing* scheme, any sentence of naturally occurring **unrestricted** English, from as wide a range of genres as possible”.

foram escritas, e diversos *parsers* já foram desenvolvidos ao longo desse tempo. Nenhum deles, porém, foi ainda capaz de alcançar o objetivo proposto pelos autores. Por isso, esse ainda continua sendo um dos principais objetivos na agenda da área de PLN, de maneira geral, e do processamento sintático das línguas naturais, de maneira específica.

Sabemos que os estudos formais da linguagem fizeram grandes avanços no século XX, mas não há ainda regras suficientemente **matematizadas**, formais, explícitas e **gerativas** que possibilitem o desenvolvimento de um *parser* que, de certa forma, “compreenda” alguma língua natural. O trabalho de descrição sintática do português que apresentamos aqui certamente pretende revelar alguns *insights* sobre a estrutura sintática das sentenças em PB e formalizar outros *insights* que alguns estudos sintáticos pioneiros já revelaram.

Por que será que ainda não existem descrições linguísticas tão acuradas, formais e matematizadas que nos permitam desenvolver *parsers* que respondam ao desafio proposto por Bateman, Forrest & Willis que vimos acima?

Um pouco de história pode nos auxiliar com a compreensão do problema. Uma das maiores decepções dos linguistas e dos cientistas da computação das décadas de 1950 e 1960 foi justamente esta: ao tentar fazer com que um computador “compreendesse” a linguagem humana, passou-se a perceber que ela é de difícil formalização. McDonald & Yazdani (1990: 6) têm uma frase muito interessante: “o grande problema em PLN é (...) que nós ainda não compreendemos completamente o funcionamento da linguagem humana”<sup>14</sup>. A frase é interessante especialmente para o linguista, cujo trabalho é justamente **compreender (completamente?) o funcionamento da linguagem humana**.

De acordo com Newmeyer (1980: 2),

houve um sentimento generalizado entre os lingüistas nos anos 1950 de que os problemas fundamentais da análise lingüística haviam sido solucionados e tudo o que restava era preencher alguns detalhes. As afirmações teórico-metodológicas básicas do trabalho de Bloch, “A Set of Postulates for Phonemic Analysis” (1948), e do trabalho de Harris, *Methods in Structural Linguistics* (1951), pareciam tornar qualquer trabalho teórico mais básico algo desnecessário. Na verdade, **muitos lingüistas achavam que os procedimentos tinham sido trabalhados de maneira tão detalhada, que computadores poderiam assumir o trabalho de análise lingüística. Tudo o que se precisava fazer (em teoria) seria colocar os dados no computador e uma gramática sairia prontinha!**

---

<sup>14</sup> Trecho original: “The major problem in NLP, however, is that we still do not fully understand the human natural language process”.

Também se acreditava que os computadores poderiam resolver outro problema lingüístico tradicional – a tradução. A idéia de uma máquina tradutora foi primeiramente sugerida apenas em 1949 (em um memorando de Warren Weaver). Em 1955, tal trabalho de tradução estava acontecendo em três países, em meia dúzia de instituições. Esses seis anos foram o suficiente para converter os cétricos (...).<sup>15</sup> (grifos nossos)

Mesmo um trabalho que parecia simples *a priori*, como a tradução automática de sentenças de uma língua para outra, mostrou-se um empreendimento extremamente complexo, dada a riqueza da linguagem. Este trecho de Baker (2001: 3) também é bastante ilustrativo e nos parece interessante:

Nós geralmente achamos que o xadrez é uma atividade intelectual requintada que pode ser dominada apenas pelos melhores e mais brilhantes. Qualquer pessoa comum, no entanto, pode falar um horror em um vernáculo entendível sem necessariamente ser considerada inteligente. Ainda assim, mesmo que existam programas de computador que agora podem derrotar os melhores jogadores de xadrez do mundo, não existe nenhum sistema artificial que possa se igualar a um falante médio de cinco anos de idade no que diz respeito a falar e compreender sua língua materna.<sup>16</sup>

De fato, já temos campeões de xadrez que são computadores<sup>17</sup>. No entanto, ainda não temos nenhum programa de computador que consiga “compreender” ou “produzir” linguagem tão bem como um ser humano<sup>18</sup>. No entanto, os estudos formais da linguagem têm contribuído para o avanço da Linguística Computacional. Alguns estudos em sintaxe formal, especialmente de cunho gerativo, já avançaram muito no conhecimento do

---

<sup>15</sup> Trecho original: “There was a widespread feeling among linguists in the 1950s that the fundamental problems of linguistic analysis had been solved and that all that was left was to fill in the details. The basic-theoretical-methodological statements of Bloch’s “A Set of Postulates for Phonemic Analysis” (1948) and Harris’s *Methods in Structural Linguistics* (1951) seemed to render any more basic theoretical work unnecessary. In fact, many linguists felt that the procedures had been so well worked out that computers could take over the drudgery of linguistic analysis. All one would have to do (in principle) would be to punch the data into the computer and out would come the grammar.”

There was also a feeling that computers could solve another traditional linguistic problem – translation. The idea of machine translation had been first suggested (in a memorandum by Warren Weaver) only in 1949. By 1955, such translation work was going on in three countries at half a dozen institutions. These six years were enough to convert the skeptics (...).”

<sup>16</sup> Tradução adaptada. Cf. trecho original: “We usually think of chess as a quintessentially intellectual activity that can be mastered only by the best and brightest. Any ordinary person, in contrast, can talk your ear off in understandable English without necessarily being regarded as intelligent for doing so. Yet although computer programs can now beat the best chess players in the world, no artificial system exists that can match an average five-year-old at speaking and understanding English”.

<sup>17</sup> Baker provavelmente está se referindo ao Deep Blue, uma máquina de jogar xadrez produzida pela IBM, que conseguiu vencer o então campeão mundial Gary Kasparov, em 1997.

<sup>18</sup> Há uma espécie de competição em que vários programas de conversação, conhecidos como *chatterbots*, tentam se passar por seres humanos. É o Prêmio Loebner, que foi baseado a partir de ideias propostas pelo teste de Turing (cf. Turing, 1950, Hodges, 1983, Oppy & Dowe, 2005).

funcionamento da linguagem humana, e muitos linguistas computacionais têm se beneficiado desses avanços. Para Newmeyer (1980: 250), “aprendemos mais sobre a natureza da linguagem nos últimos 25 anos do que nos 2.500 anos anteriores”<sup>19</sup>. Carnie (2002: 5), por exemplo, afirma que

a tese subjacente da gramática gerativa é a de que as sentenças são geradas por um conjunto subconsciente de procedimentos (como programas de computador). Esses procedimentos são parte de nossas mentes (...). O objetivo da teoria sintática é modelar esses procedimentos.<sup>20</sup> (grifos nossos).

Acreditamos que o ser humano tenha um conhecimento inato do funcionamento sintático de sua língua. Há regras a que os falantes obedecem ao produzirem os enunciados, regras “subconscientes” que os falantes não sabem explicitar. O trabalho em sintaxe formal e computacional pode ser entendido desta maneira então: deve-se explicitar a um computador as regras de funcionamento sintático da língua para que o computador possa então “compreender” a linguagem. Um programa como um *parser*, que apresentaremos no capítulo 3, analisa uma sentença em linguagem natural (o português, por exemplo), dizendo se a sentença é gramatical ou agramatical na língua. Além disso, ele atribui às sentenças gramaticais a sua correta estrutura sintática (de acordo com a gramática que foi implementada no programa). Para isso, é necessário o trabalho prévio do linguista: é ele quem deve formalizar as regras sintáticas de uma língua de tal maneira que elas possam ser implementadas e tratadas computacionalmente. Então, uma estratégia para o desenvolvimento de um *parser* seria investi-lo do conhecimento já explicitado sobre os processos de formação das sentenças e dos sintagmas da língua. Ou seja: deveríamos ensinar ao *parser* tudo aquilo que as gramáticas sabem. Evidentemente, seria preciso antes formalizar as regras gramaticais para que elas pudessem ser manipuladas pelo computador. E aí entra este nosso trabalho: tentaremos propor regras formais de descrição sintática do português que deem conta da análise da estrutura dos sintagmas e de uma vasta gama de sentenças em PB.

---

<sup>19</sup> Trecho original: “(...) more has been learned about the nature of language in the last 25 years than in the previous 2500”. Essa parece ser uma ideia mais ou menos generalizada entre os linguistas gerativos. Para Strozer (1994: ix), os estudos linguísticos “avançaram mais nos últimos dez ou quinze anos do que nos 30 anos anteriores”. Jairo Nunes (em comunicação pessoal) disse que frases como essas são comuns em seções introdutórias de textos recentes dos estudos gerativos. Cf. também a esse respeito Harris (1993), Chomsky (1988, 1995, 2002) e Thomas (2004).

<sup>20</sup> Trecho original: “The underlying thesis of generative grammar is that sentences are generated by a subconscious set of procedures (like computer programs). These procedures are part of our minds (...). The goal of syntactic theory is to model these procedures”.

Certamente nossas regras não serão capazes de descrever *todas* as estruturas dos sintagmas em português. Teremos algum mérito se as regras que apresentarmos conseguirem prever apenas estruturas bem formadas e dar conta de uma ampla variedade de sintagmas. Veremos que essa é uma tarefa complicada já em sua natureza. Mas contamos com o auxílio de estudos pioneiros da sintaxe do português para propor novas regras categoriais de reescrita que consigam analisar corretamente algumas estruturas da sentença em PB. Ao tentar propor uma gramática do português com regras de reescrita, lembramos uma passagem de Archangeli & Langendoen (1997: viii), que comparam a tarefa do linguista com a tarefa de um pescador:

Os lingüistas são, então, confrontados com dois problemas relacionados. Um é garantir que a gramática de uma língua particular seja capaz de abranger todas as expressões que podem pertencer à língua. A outra é garantir que a gramática seja capaz de distinguir aquelas expressões que pertencem à língua daquelas que não pertencem à língua. O problema pode ser comparado ao problema de um pescador tentando pegar em sua rede todos os peixes de um determinado tipo em uma certa região, mas nada mais (nenhum peixe de outra espécie, nenhuma outra criatura que não seja peixe, etc.). A rede ideal deverá ser grande e fina o suficiente para pegar todos os peixes desejados pelo pescador (os desejáveis), e deverá ser projetada para permitir que os peixes indesejados e outras criaturas (os indesejáveis) possam escapar. Mas pode não ser possível criar este tipo de rede. Qualquer rede que seja grande e fina o suficiente para apanhar todos os desejáveis pode precisar também apanhar alguns indesejáveis<sup>21</sup>.

Esperamos que o leitor lembre-se do problema e da complexidade da tarefa de descrição gramatical ao chegar ao capítulo 4, quando apresentaremos nossas regras categoriais de reescrita para a descrição dos sintagmas e da sentença em PB. No próximo capítulo, apresentaremos o modelo sintático que iremos utilizar em nossas descrições do português, a teoria X-barra.

---

<sup>21</sup> Trecho original: “Linguists are thus faced with two related problems. One is to ensure that the grammar of a particular language is able to encompass all of the expressions that can reasonably be supposed to belong to that language. The other is to ensure that the grammar is able to distinguish those expressions which belong to the language from those which do not.

The problem can be compared to that of a fisherman trying to catch in a net all the fish of certain types in a certain area, but nothing else (no other types of fish, no other creatures, etc.). The ideal net would be large and fine enough to gather all the desired fish (the desirables), and be designed to allow the undesired fish and other creatures (the undesirables) to escape. But it may not be possible to construct such a net. Any net which is large and fine enough to catch all the desirables may of necessity also catch some undesirables”.

## 2 O FORMALISMO GRAMATICAL

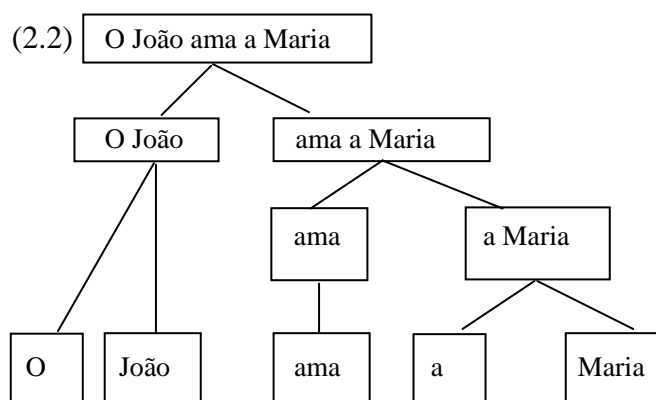
*Fornecer definições formais de noções importantes faz com que seja mais fácil achar inconsistências e insuficiências na teoria. Em teorias descritas informalmente é difícil achar problemas. (...) É praticamente impossível determinar as conseqüências de teorias informais e, assim, de mostrá-las falsas. Teorias formalizadas, por sua vez, levam à descoberta de inconsistências e a soluções para problemas outros, além daqueles para os quais a teoria foi originalmente desenhada para tratar. Quando usamos uma linguagem formal, somos forçados a ser mais precisos; todas as noções usadas em uma definição devem ser definidas elas mesmas.*

Marcello Modesto

### 2.1 A GRAMÁTICA SINTAGMÁTICA

Ao efetuarmos análises das sentenças em português, estaremos pressupondo que elas apresentem uma determinada organização sintática **estrutural**. Por exemplo, ao analisarmos (2.1), iremos fazer algo como (2.2):

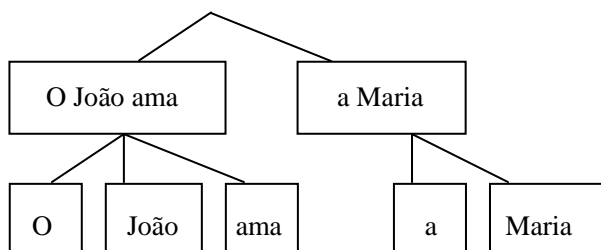
(2.1) O João ama a Maria.



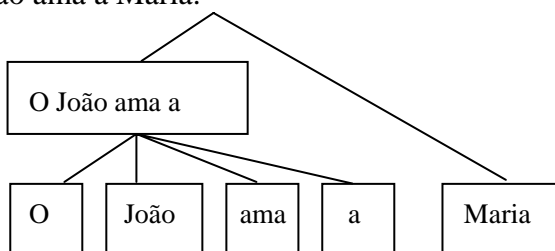
Não analisamos a sentença (2.1) sem nenhum critério, agrupando quaisquer grupos de palavras aleatoriamente. Se assim fosse, poderíamos tê-la analisado de várias outras maneiras, formando qualquer tipo de agrupamento interno na frase, até chegarmos aos itens lexicais, como em (2.3) e (2.4). Poderíamos até mesmo supor que a sentença não

tenha qualquer organização interna, sendo formada simplesmente por um aglomerado linear de palavras, como em (2.5):

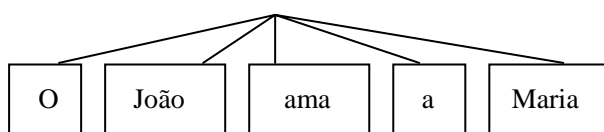
(2.3) O João ama a Maria.



(2.4) O João ama a Maria.



(2.5) O João ama a Maria.



Não iremos encarar a sentença como um mero aglomerado de palavras, unidas uma a outra de qualquer forma. Há entre o nível da palavra e o da frase uma outra forma de organização, que é o sintagma (ou constituinte). De acordo com Radford (1981: 69),

um certo agrupamento de palavras é um constituinte somente se tiver uma ou mais das seguintes propriedades:

- (i) Comporta-se distribucionalmente como uma única unidade estrutural, isto é, é recorrente como uma unidade única em uma variedade de outras posições nas frases
- (ii) Pode ser coordenado com outro agrupamento similar
- (iii) Não permite intrusão interna de elementos parentéticos (a intrusão geralmente sendo permitida apenas nas fronteiras de constituintes maiores, especialmente sintagmas)
- (iv) Pode ser substituído por (ou servir de antecedente para) uma proforma
- (v) Pode ser omitido, sob condições de discurso apropriadas<sup>22</sup>.

<sup>22</sup> Trecho original: "A given string of elements is a constituent just in case it has one or more of the following properties:

(i) It behaves distributionally as a single structural unit – i. e. it recurs as a single unit in a variety of other

Isso quer dizer que podemos realizar diferentes “testes” para identificarmos um constituinte, como os testes de coordenação, interpolação, anáfora, entre outros<sup>23</sup>. Esses testes permitirão identificar os limites dos constituintes, confirmando ou não nossa própria intuição linguística. Essa maneira de “enxergar” e analisar a sentença e seus constituintes é conhecida como *Phrase Structure Grammar*, ou PSG.

A PSG é um formalismo de base gerativa referido por Chomsky (especialmente em 1955, 1956 e 1957) que, nos anos 1980, teve um grande impulso graças a estudos sintáticos formais não transformacionais bastante interessantes, como a *Generalized Phrase Structure Grammar* (GPSG) e, mais tarde, a *Head-driven Phrase Structure Grammar* (HPSG)<sup>24</sup>. Uma PSG possui quatro componentes centrais: (i) um conjunto finito de símbolos terminais; (ii) um conjunto finito de símbolos não terminais; (iii) um conjunto finito de regras gramaticais; e (iv) um símbolo inicial.

Em regras como

$$(2.6) \quad S \rightarrow A B$$

$$(2.7) \quad A \rightarrow a$$

$$(2.8) \quad B \rightarrow b$$

temos um símbolo inicial **S**, símbolos não terminais **A** e **B** e símbolos terminais **a** e **b**. Há diferentes maneiras de se entender as regras acima<sup>25</sup>; entenderemos as regras de reescrita como instruções para um *parser top-down*, ou seja, um mecanismo de construção de estruturas que começa seu funcionamento de cima para baixo. De acordo com a regra (2.6), **S** pode ser expandido em **A** e **B**; seguindo a regra (2.7), **A** pode ser expandido em **a**; finalmente, a regra (2.8) diz que **B** pode ser expandido em **b**. Como resultado, obtemos a seguinte estrutura sintagmática:

---

sentence positions

(ii) It can be coordinated with another similar string

(iii) It does not readily permit intrusion of parenthetical elements internally (intrusion generally being permitted only at the boundaries of major – especially phrasal – constituents)

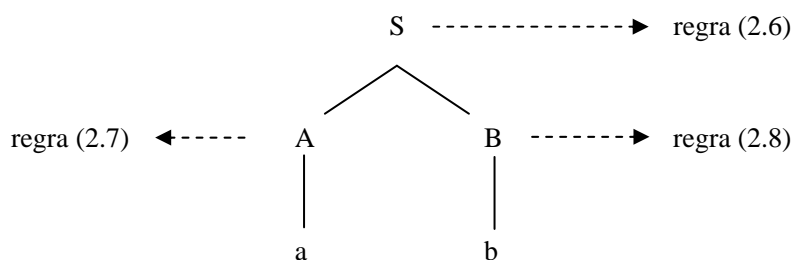
(iv) It can be replaced by, or serve as the antecedent of, a proform

(v) It can be omitted, under appropriate discourse conditions”

<sup>23</sup> Ver a esse respeito Radford (1981, 1988), Aarts & Haegeman (2006) e Perini (2006).

<sup>24</sup> Sobre a GPSG, cf. Gazdar et al. (1985); sobre a HPSG, cf. Pollard & Sag (1987, 1994).

<sup>25</sup> Cf. Borsley (1999), cap. 2.



Há dois pontos importantes que devemos notar: (i) as regras de uma PSG são **gerativas**, ou seja, são explicitamente descritas; e (ii) elas apresentam um símbolo não terminal à esquerda e um símbolo (ou uma sequência de símbolos) terminal à direita. Aqui estamos nos referindo a uma gramática PSG **livre de contexto** (CFG, do inglês *context-free grammar*). Uma PSG livre de contexto é

um tipo específico de sistema formal que tem se mostrado bastante útil em precisar a caracterização de línguas computacionais e que também serve como ponto inicial para muitos trabalhos em teoria sintática. As CFGs consistem em um símbolo inicial, um léxico finito com palavras classificadas em categorias gramaticais e um conjunto finito de regras da forma  $A \rightarrow \omega$ , onde  $A$  é um símbolo único (representando um tipo de sintagma) e  $\omega$  é uma seqüência finita de categorias lexicais e/ou sintagmáticas.<sup>26</sup> (Sag, Wasow & Bender, 2003: 558)

Ainda de acordo com Sag, Wasow & Bender (2003: 26), as CFGs são “o ponto inicial para a maioria das tentativas sérias de se desenvolver gramáticas formais para as línguas naturais”. Soames & Perlmutter (1979: 26) vão mais longe e afirmam que “as regras livres de contexto são o único tipo de regras de estrutura sintagmática de que se precisa nas gramáticas das línguas naturais”<sup>27</sup>.

Todas as regras de boa formação da sentença e dos sintagmas em PB que apresentaremos seguirão o estilo de regras de uma gramática de estrutura sintagmática livre de contexto. Na verdade, utilizaremos uma “instância” da PSG, a saber, o esquema proposto pela **teoria X-barra**.

<sup>26</sup> Trecho original: “A context-free grammar is a particular type of formal system that has proved very useful in the precise characterization of computer languages and also serves as the starting point for much work in syntactic theory. CFGs consist of an initial symbol, a finite lexicon with words classified into grammatical categories, and a finite collection of rules of the form  $A \rightarrow \omega$ , where  $A$  is a single symbol (representing a type of phrase), and  $\omega$  is a finite string of lexical and/or phrasal category”.

<sup>27</sup> Trechos originais: “(...) the starting point for most serious attempts to develop formal grammars for natural languages” (Sag, Wasow & Bender, 2003: 26); “(...) context-free rules are the only kind of phrase structure rules needed in the grammar of natural languages” (Soames & Perlmutter, 1979: 26). Ver, no entanto, Chomsky (1956), Pullum & Gazdar (1982), Shieber (1985), Savitch et al. (1987), Wong & Wu (1999) e Hopcroft et al. (2001), para críticas às regras de PSGs livres de contexto e sua utilização no trabalho de descrição linguística. Uma boa introdução ao assunto pode ser vista em Gazdar (1983).

## 2.2 TEORIA X-BARRA

A teoria X-barra é o modelo que iremos utilizar para dar conta da organização interna dos sintagmas e da sentença. De acordo com Alencar (no prelo),

a teoria X-barra é um dos pilares do modelo Princípios e Parâmetros da lingüística gerativa. Trata-se de uma concepção restritiva da gramática de estrutura sintagmática, a qual permite análises psicolinguisticamente mais realistas e computacionalmente mais elegantes do que as praticadas anteriormente no âmbito da gramática gerativa. Desse modo, essa teoria tem sido empregada também na lingüística computacional, em análises baseadas em formalismos gramaticais de natureza não transformacional que operam com a unificação de traços.

No entanto, a teoria X-barra não é restrita ao estudo sintático em P&P e GB. A teoria X-barra é, na verdade, uma teoria sobre a estrutura das sentenças e dos constituintes nas línguas naturais. Ela é uma teoria adotada por diferentes modelos de descrição linguística. Como apontam Kornai & Pullum (1990: 2), a teoria X-barra é

discutida em quase todos os livros modernos de sintaxe, e é rotineiramente aceita como uma teoria da estrutura sintagmática em uma variedade de escolas distintas de pensamento gramatical, como a Teoria da Regência e da Ligação (GB), *Lexical-Functional Grammar* (LFG) e *Generalized Phrase Structure Grammar* (GPSG).<sup>28</sup>

Um dos motivos pelos quais resolvemos adotar o modelo da teoria X-barra para nossas descrições da sintaxe do PB foi justamente este: a teoria X-barra é uma teoria sintática coerente que já está no mercado há mais de trinta anos<sup>29</sup> e tem rendido estudos interessantes nas línguas naturais. Além do mais, conhecemos apenas uma proposta abrangente de descrição sintática da estrutura da sentença do português seguindo o modelo, a proposta de Lemle (1984)<sup>30</sup>. Aqui, em nosso texto, pretendemos rever as

---

<sup>28</sup> Trecho original: “(...) is discussed in almost all modern textbooks of syntax, and it is routinely assumed as a theory of phrase structure in a variety of otherwise widely differing schools of grammatical thought such as government-binding theory (GB), lexical-functional grammar (LFG), and generalized phrase structure grammar (GPSG)”. Além disso, o modelo proposto pela teoria X-barra também é adotado em trabalhos em **Teoria da Otimidade** (cf. Legendre, 2001, Kuhn, 2003, entre outros), é parcialmente utilizado pela proposta do *Simpler Syntax* (cf. Culicover & Jackendoff, 2005, cf. especialmente p. 22) e pelo formalismo da **HPSG** (cf. Pollard & Sag, 1994, e Sag, Wasow & Bender, 2003).

<sup>29</sup> As bases da teoria X-barra remetem aos trabalhos de Chomsky (1970), Jackendoff (1977) e Stowell (1981).

<sup>30</sup> Evidentemente, há outros pesquisadores que utilizam a teoria X-barra para analisar fragmentos do português, ou introduzir questões de análise do português, como Lobato (1986), Raposo (1992), Costa (1998), Miotto et al. (2004), etc.

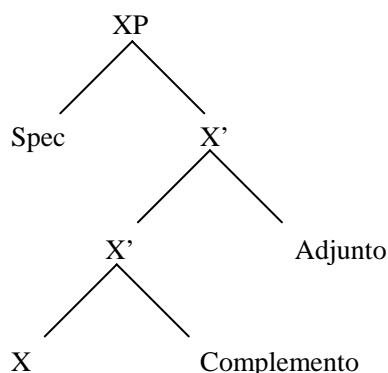
descrições de Lemle e atualizá-las ao modelo corrente de análise sintática no modelo da X-barra. Além disso, existem trabalhos de descrição sintática do português bastante interessantes (como Pontes, 1973; Lobato, 1975, 1986; Luft, 1986; Perini, 1989, 2000) que não foram devidamente formalizados e não se utilizam do modelo teórico da X-barra. Talvez a formalização desses estudos seguindo o modelo da teoria X-barra possa revelar aspectos interessantes, tanto para o modelo como para a descrição sintática do português.

Por fim, acreditamos que a formalização de tais descrições do português no modelo X-barra poderá nos fornecer uma gramática passível de implementação computacional. Sabemos que a teoria ainda não foi devidamente trabalhada em termos computacionais e ainda pode revelar potencial. Há alguns trabalhos interessantes para a língua inglesa (Chen & Wu, 1988; Black, 1997; Fong, 1999, 2001; Fouskakis, 2005; entre outros) e também para a língua portuguesa (Pagani, 2004; David, 2007; Alencar, no prelo), incluindo nossos próprios trabalhos, como Othero (2004, 2006), Menuzzi & Othero (2008) e Othero & David (no prelo). Esperamos dar uma contribuição para os estudos gramaticais em língua portuguesa ao propor nossa descrição da sentença e de seus constituintes seguindo o molde da X-barra.

Para Mioto et al. (2004: 46), a teoria X-barra é

o módulo da gramática que permite representar um constituinte. Ela é necessária para explicitar a natureza do constituinte, as relações que se estabelecem dentro dele e o modo como os constituintes se hierarquizam para formar a sentença.

Em outras palavras, a teoria X-barra trata sobre a organização sintática **estrutural** dos constituintes da sentença. Ela apresenta um “molde”, um **modelo** para a descrição sintática dos tipos de sintagmas nas línguas naturais. Para nossas análises, partiremos do seguinte molde para a descrição da estrutura arbórea dos constituintes:



Onde temos a projeção máxima de um sintagma **XP** (ou **X''**), a projeção intermediária **X'** e seu núcleo **X** (ou **X<sup>0</sup>**). Há ainda três outras posições: **Spec**, **Adjunto** e **Complemento**. O complemento está mais próximo ao núcleo (é **irmão** de X), enquanto o adjunto vem ao lado do constituinte intermediário (é **irmão** de X')<sup>31</sup>. Repare que essa é uma definição **estrutural** de complementos e adjuntos (ela não leva em consideração detalhes de qualquer outra natureza que não esteja refletida na estrutura da sentença – como as propriedades semânticas dos constituintes, por exemplo). Essa estrutura arbórea permite refletir o fato de que os complementos costumam estar mais próximos ao núcleo do que os adjuntos<sup>32</sup>. A posição de **Spec** será ocupada pelo sujeito da sentença ou por algum outro elemento modificador externo.

A representação dos constituintes em estrutura X-barras traz algumas vantagens em relação a uma simples PSG. Como vimos no esquema acima, o modelo X-barras prevê corretamente uma organização hierárquica que seja maior do que o X e menor do que o XP, além de fazer uma distinção sintática entre complementos e adjuntos<sup>33</sup>.

Assumiremos o esquema da teoria X-barras que apresentamos acima, juntamente com alguns princípios da teoria<sup>34</sup>:

- a) Endocentricidade
- b) Lexicalidade
- c) Sucessão
- d) Uniformidade
- e) Maximalidade

---

<sup>31</sup> Essa é **uma** maneira de trabalhar com o modelo da teoria X-barras, seguindo, por exemplo, Lobato (1986), Haegeman (1995), Culicover (1997), Radford (1997a), Carnie (2002), Santorini & Kroch (2007), entre outros. Mas ela não é consensual. Há autores, como Chomsky (1986b), Miotto et al. (2004) e Cançado (2005), que tratam os adjuntos como filhos da projeção máxima XP e não da projeção intermediária X'.

<sup>32</sup> A bibliografia que argumenta em favor dessa ideia é vasta. Sérgio Menuzzi (em comunicação pessoal), diz que representar um complemento próximo de seu núcleo na estrutura arbórea sugere que as relações de dominância sintática refletem iconicamente o grau de “intimidade semântica” do modificador em relação ao núcleo. Em línguas VO, como o português, os adjuntos costumam vir em posição final (VO+Adjuntos). Em línguas OV, como o alemão, os adjuntos deixam sua posição final e passam a ocupar a posição anterior aos complementos (Adjuntos+OV). Isso revela uma generalização interessante: os núcleos verbais e seus argumentos internos parecem formar um constituinte coeso, deixando elementos modificadores ocuparem uma posição satélite.

<sup>33</sup> Motivações empíricas para se assumir esses dois pontos são apresentadas largamente em Radford (1988) e Raposo (1992).

<sup>34</sup> Iremos nos basear especialmente nos princípios discutidos no artigo clássico de Kornai & Pullum (1990). Deixemos claro que esses princípios não são aceitos unanimemente; cada um deles já foi desafiado – cf., por exemplo, o próprio texto de Kornai & Pullum (1990), além de Kayne (1984, 1994) e Culicover & Jackendoff (2005).

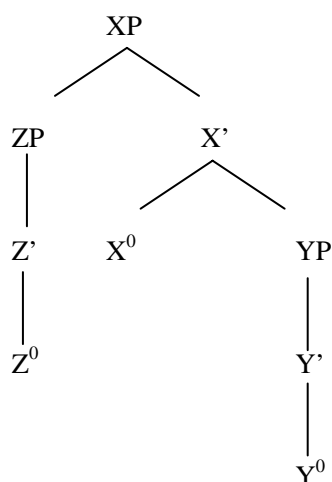
- f) Opcionalidade
- g) Binaridade

A **endocentricidade** é o princípio que garante que cada constituinte **XP** tenha o núcleo **X** (e não um núcleo **Y** ou **Z**):



Todos os sintagmas são encabeçados por um núcleo (uma projeção zero,  $X^0$ ), de mesma natureza que a projeção máxima.

O princípio da **lexicalidade** postula que cada elemento não terminal seja a projeção de um elemento terminal, como mostra o diagrama abaixo:



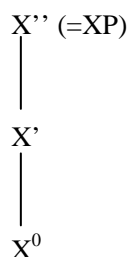
Juntamente com o princípio da endocentricidade, garantimos que cada elemento **XP** ou **X'** tenha seu núcleo  $X^0$ . Esse não foi um princípio respeitado em Lemle (1984), Lobato (1986) e Othero (2004), por exemplo, que seguiram propostas mais antigas da teoria X-barra, que ainda contavam com um elemento não terminal **S**, da sentença, que não possuía qualquer núcleo. Esse elemento se ramificava em dois outros elementos não terminais, **NP** e **VP**, por exemplo, não apresentando, portanto, um núcleo terminal, ou lexical,  $X^0$ , o que acabou desrespeitando também o princípio da endocentricidade

(voltaremos ao assunto em seguida).

O princípio de **sucessão** diz respeito à relação de dominância entre os elementos constituintes do sintagma. De acordo com esse princípio, cada  $X^{n+1}$  domina  $X^n$ , onde  $n \geq 0$ . Na verdade, iremos respeitar apenas parcialmente o princípio de sucessão, uma vez que iremos propor (seguindo Lobato, 1986; Miotto et al., 2004; e Raposo, 1992, entre outros) que um elemento  $X^n$  pode dominar um outro elemento  $X^n$  (isso acontecerá quando tivermos sintagmas verbais com mais de um complemento, ou com mais de um modificador adverbial, por exemplo). Adotaremos o esquema

$$X^n \rightarrow \dots X^m \dots \text{ (onde } m = n \text{ ou } n-1 \text{)}.$$

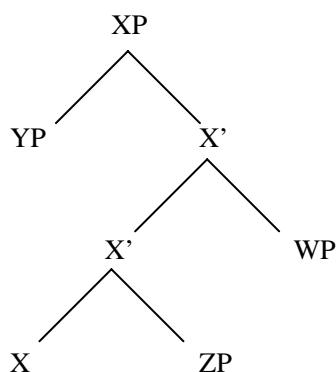
O princípio da **uniformidade** garante que todas as projeções máximas tenham o mesmo número de barras. Em nossas análises, o valor de **n** máximo de barras será igual a 2 (que é o número mais usualmente adotado para o número máximo de barras em uma projeção máxima, cf. Radford, 1981, 1988; Lobato, 1986; Raposo, 1992, entre outros):



Apesar de haver mais ou menos um consenso de que o número máximo de barras seja igual a 2, este também não é um ponto pacífico entre os sintaticistas. No texto clássico de Jackendoff (1977), em que ele descreve a estrutura do inglês de acordo com o modelo X-barra, o número máximo de barras é 3. No trabalho de Lemle (1984), a autora assumia como número máximo de barras ora 2, ora 3, ora 4, sem estabelecer um *n* máximo limite. Outros estudos importantes que se utilizaram da teoria X-barra, como Williams (1975), Emmonds (1976) e Stuurman (1985), postularam um número máximo diferente, ou sequer respeitaram o princípio da uniformidade, apresentando projeções máximas de diferentes núcleos com diferentes números máximos de barra.

**Maximalidade** é o princípio que postula que todos os elementos funcionais também devem ser projeções máximas. Esse princípio é talvez o mais controverso, já que

pode “complicar” a análise e a descrição sintática de estruturas sintagmáticas de superfície<sup>35</sup>. Alguns estudos para o português, como Lemle (1984), Lobato (1986), Souza e Silva & Koch (1993) e Othero (2006), não seguiram esse princípio e utilizaram rótulos como “determinante”, “pré-determinante”, etc., que não obedeciam ao modelo X-barra. Ou seja, esses elementos não apresentavam a estrutura interna XP – X’ – X<sup>0</sup>. De acordo com **maximalidade**, devemos entender o esquema X-barra que apresentamos acima como o seguinte:



Onde XP, YP, WP e ZP são projeções.

Por fim, **binaridade** é um princípio que regula a organização esquemática da X-barra. Repare que as ramificações estruturais na árvore sintática são sempre binárias, o que irá acarretar que nossas regras de reescrita também deverão ser sempre binárias. Para Santorini & Kroch (2007: 70), “hoje é largamente aceita a suposição de que a estrutura sintática é, no máximo, ramificada binariamente (em outras palavras a ramificação binária parece ser uma forma universal)<sup>36</sup>”. No entanto, há casos em que a binaridade não é clara, como em estruturas coordenadas (**[João e Maria] foram ao cinema**) e na representação de verbos com mais de um argumento semântico (**João [deu-me um presente]**). Voltaremos ao assunto adiante<sup>37</sup>.

<sup>35</sup> Ver, por exemplo, as críticas de Culicover & Jackendoff (2005).

<sup>36</sup> Trecho original: “it is now widely assumed that syntactic structure is at most binary-branching (in other words, binary-branchingness is assumed to be a formal universal)”.

<sup>37</sup> Ainda sobre a binaridade, cf. Kayne (1984) e Kornai & Pullum (1990).

## 2.3 AS PROJEÇÕES MÁXIMAS EM NOSSAS ANÁLISES

Iremos agora mostrar quais serão as projeções máximas que adotamos em nossas análises da estrutura sintagmática do português. Começemos pelos cinco núcleos lexicais: NP, VP, AP, AdvP e PP<sup>38</sup>.

Há boas evidências na literatura de que existam pelo menos estas quatro projeções máximas cujo núcleo seja um elemento lexical: NP, VP, AP e PP. Além da vasta literatura em sintaxe gerativa que adota essas quatro projeções máximas, outros modelos e formalismos em sintaxe também costumam adotar esses quatro rótulos em suas análises, desde estudos gramaticais mais “tradicionais” (Luft, 1986; e Azeredo, 2000, por exemplo) até modelos mais recentes como o *Simpler Syntax*, a HPSG e a Teoria da Otimidade.

Os núcleos lexicais são identificados pela combinação de categorias lexicais de dois traços distintivos fundamentais, nominal [N] e verbal [V], que podem ter dois valores, positivo (+) ou negativo (-). Combinando os valores, obtemos a seguinte tabela<sup>39</sup>:

	[+N]	[-N]
[-V]	Nome	Preposição
[+V]	Adjetivo	Verbo

Tabela 2.1: Núcleos lexicais

Chamamos essas projeções de lexicais porque seus núcleos podem **subcategorizar** seus complementos. Essas quatro classes gramaticais têm seu *status* relativamente garantido entre os estudos gramaticais. No entanto, repare que não há lugar para o advérbio na tabela acima, ainda que, acreditamos, o advérbio também possa subcategorizar seus complementos. Mateus & Xavier (1992: 611), por exemplo, dizem que o sintagma adverbial é a

projecção máxima de um advérbio. É, geralmente, constituído apenas pelo advérbio ou por este e um especificador também adverbial, funcionando como quantificador. O advérbio pode seleccionar um complemento. Exemplo: independentemente de tudo.

<sup>38</sup> Mioto et al. (2004) defendem a ideia de que existem **preposições lexicais** e **preposições funcionais**. Essa distinção não será relevante para nosso estudo.

<sup>39</sup> Essa classificação data de Chomsky (1970).

Lemle (1984: 130) também concorda com a postulação da projeção máxima adverbial: “um advérbio pode ter complementos, e um advérbio mais os seus complementos perfaz um sintagma adverbial”. Diversas outras propostas de descrição gramatical igualmente propõem a existência do sintagma adverbial, como Jackendoff (1977), Huddleston (1984), Quirk et al. (1985), Aarts & Haegeman (2006), etc<sup>40</sup>.

Adotamos o advérbio como núcleo de uma projeção máxima em trabalhos anteriores (Othero, 2006; Menuzzi & Othero, 2008) e continuaremos a fazê-lo aqui. Em resumo, adotaremos estes cinco núcleos lexicais, **N**, **P**, **A**, **V** e **Adv**, cada um deles com sua projeção máxima, **NP**, **PP**, **AP**, **VP** e **AdvP**.

Antes de prosseguir, seremos obrigados a fazer um *mea culpa*: não iremos nos aprofundar na definição exata de cada um desses núcleos lexicais. Acreditamos que um estudo que pretenda definir cada uma dessas classes gramaticais por si só poderia trazer trabalho suficiente para render um livro<sup>41</sup>. Por isso, seguiremos o consenso entre os estudos gramaticais do português sempre que possível e adotaremos as definições que encontramos ali. Quando lidarmos com casos problemáticos (como nas seções 4.3 e 4.4, em que trataremos do sintagma preposicional e do sintagma adverbial, respectivamente), faremos algumas escolhas, tomaremos algumas posições e argumentaremos em favor delas. O mesmo irá valer sobre os núcleos funcionais de que falaremos a seguir.

Os núcleos funcionais têm função essencialmente sintática e não subcategorizam seus complementos<sup>42</sup>. Na verdade, em nosso trabalho, a distinção entre núcleos lexicais e núcleos funcionais não será relevante. O importante é estipularmos quais tipos de núcleos e de projeções máximas iremos adotar na descrição dos sintagmas e da sentença em português e justificar sua importância com a descrição sintática que iremos apresentar.

Uma projeção máxima que utilizaremos em nossas análises será a projeção máxima do **DP**, o sintagma determinante (do inglês *determiner phrase*), projeção máxima do núcleo **determinante**. Desde o final da década de 1980, graças principalmente ao trabalho de Abney (1987), o antigo NP passou a ser entendido como uma projeção do

<sup>40</sup> Winograd (1983: 53) também “aceita” o advérbio como categoria gramatical legítima na língua (inglesa), ainda que com uma advertência: “[o advérbio] é a sobra entre as categorias (*leftover category*) – qualquer coisa que não se encaixa em outra classe recebe o nome de advérbio”!

<sup>41</sup> O assunto da categorização das classes de palavras é tratado por diversos autores – para uma introdução ao problema, ver Assunção (2001), Azeredo (2001), Perini (2003, 2006) e Aarts & Haegeman (2006); para um estudo linguístico pioneiro em português do Brasil, ver Câmara Jr. (1970); para uma tentativa bastante ampla e interessante, ver Perini (1989, 2000).

<sup>42</sup> Uma diferença interessante entre núcleos lexicais e funcionais tem caráter semântico: de acordo com Glanzberg (2008), os núcleos funcionais não podem ser interpretados metaforicamente, enquanto os núcleos lexicais podem.

determinante. Em nossa descrição do sintagma nominal, adotaremos a projeção máxima DP, aceitando, então, a **hipótese do DP**<sup>43</sup>. Assumiremos que a posição do D em português pode ser ocupada por **pronomes demonstrativos, pronomes pessoais, artigos, quantificadores, elementos-QU** ou por um **elemento vazio** (entraremos em mais detalhes na seção 4.1).

Ainda na descrição do sintagma nominal, adotaremos os núcleos **Q** (quantificador), **Poss** (possessivo) e **Num** (numeral), que projetam **QP, PossP** e **NumP**, respectivamente (cf. 4.1). Nossas propostas de análises dos agrupamentos nominais do português se baseiam boa parte em Abney (1987), Radford (1997a,b) e Zamparelli (2000), que apresentam análises para o inglês; e Lemle (1984), Luft (1986), Perini (1989, 2000), Raposo (1992) e Miotto et al. (2004), com suas análises para o português. Seguimos de perto também a proposta de David (2007) e Othero & David (no prelo), que descrevem a estrutura do DP visando à implementação computacional.

Para o estudo da estrutura e descrição da **sentença**, ao contrário do que fizemos em trabalhos anteriores, não iremos utilizar o símbolo **S**. Como vimos anteriormente, uma regra tal como (2.9) violaria alguns princípios da teoria X-barra, como o princípio da endocentricidade:

$$(2.9) S \rightarrow NP VP^{44}$$

Para a descrição da sentença em português, adotaremos as duas projeções máximas **IP** (sintagma flexional, do inglês *inflectional phrase*); e **CP** (sintagma complementizador, do inglês *complementizer phrase*), seguindo modelos correntes de descrição gramatical que também utilizam o formalismo da teoria X-barra. Um breve histórico da evolução do estudo da sentença, de **S** até as projeções **IP** e **CP**, pode ser vista em Falk (2006), de quem tomamos emprestados alguns trechos:

A sentença como uma categoria sui generis. Em sintaxe gerativa, a categoria S é analisada expandindo-se como uma seqüência de NP-VP

---

<sup>43</sup> Cf. Abney (1987) e Coene & D'Hulst (2003a, 2003b), entre outros. As análises com o DP são mais ou menos consensuais em pesquisa sintática atual, dentro da Teoria dos Princípios e Parâmetros (cf., no entanto, Cinque, 1995, e Escribano, 2006, para críticas à hipótese do DP).

<sup>44</sup> Tal regra é comumente encontrada em manuais de sintaxe antigos para a descrição da sentença em português. Nós mesmos implementamos computacionalmente essa regra na gramática do *parser* que apresentaremos no próximo capítulo, o Grammar Play. Na verdade, encarar a sentença como o símbolo inicial **S** era uma prática comum até Chomsky (1986b), que substitui S' por CP (*Complementizer Phrase*) e S por IP (*Inflectional Phrase*), cada um com sua estrutura interna respeitando lexicalidade e endocentricidade (IP – I' – I(nfl); CP – C' – C(omp)), como veremos em seguida.

(...).

Chomsky (1957) caracteriza o componente de regras sintagmáticas como sendo “definido por um conjunto finito  $\Sigma$  de seqüências iniciais e um conjunto finito F de ‘fórmulas de instrução’ com a forma  $X \rightarrow Y...$ ” [Chomsky, 1957: 29], e descreve as gramáticas sintagmáticas como gramáticas  $[\Sigma, F]$ . O conceito do símbolo inicial é então considerado parte da definição do componente sintagmático da gramática. O símbolo inicial é a sentença, como se percebe pela discussão sobre se o símbolo  $\Sigma$  deve consistir apenas pela Sentença ou se devem ser incluídas categorias distintas como Sentença Declarativa, Sentença Interrogativa, etc. Uma outra indicação do status especial da sentença/oração nesse trabalho é o tratamento da recursão. Enquanto a recursão era geralmente codificada nas regras de estrutura sintagmática, a recursão oracional era analisada pela geração de sentenças separadas e então encaixadas uma na outra por uma transformação generalizada.

Em Chomsky (1965), o uso de um mecanismo formal distinto para a recursão frasal foi abandonado, mas a singularidade da sentença foi mantida.

Chomsky (1970), que introduziu a teoria X’, declarou que a ‘regra inicial’ da base era  $S \rightarrow N'' V''$ , sem nenhuma proposta de incorporar a sentença ao sistema X-barra. (...)

Jackendoff (1977), em sua tentativa de generalizar estruturalmente o sujeito-da-sentença com o possuidor-do-NP (assumindo que o possuidor fosse especificador do NP), propõe que uma sentença seja a projeção máxima do V. VP é uma categoria submáxima, isto é, não é uma oração plena. (Os auxiliares também são especificadores).

Gazdar, Pullum & Sag (1982) apresentam uma análise semelhante à de Jackendoff (sentença = V’, “VP” = V’). A única diferença é que eles analisam auxiliares como sendo verbos. (...)

A sentença como projeção máxima da Inflexão (I). Chomsky (1981) menciona essa possibilidade. (...)

Chomsky (1986b) popularizou essa análise (...). Essa implementação tem a vantagem de fornecer uma estrutura X’ plena.<sup>45</sup>

---

<sup>45</sup> Trecho original: “Sentence is a sui generis category. In generative syntax this has been realized as analyzing the special category S as expanding as a subject-predicate sequence of NP-VP (...).

Chomsky (1957) characterizes the phrase structure rule component of the grammar as being ‘defined by a finite set  $\Sigma$  of initial strings and a finite set F of ‘instruction formulas’ of the form  $X \rightarrow Y...$ ’, and describes phrase structure grammars as  $[\Sigma, F]$  grammars. The concept of an initial symbol is thus taken to be part of the definition of the phrase structure component of the grammar. This initial symbol is the sentence, as is made clear by the discussion over whether the set  $\Sigma$  consists just of Sentence or might include distinct categories Declarative Sentence, Interrogative Sentence, etc. Another indication of the special status of the sentence/clause in this early work is the treatment of recursion. While recursion was generally encoded in the phrase structure rules, clausal recursion was analyzed by generating separate sentences, and then embedding one in the other by a generalized transformation.

In Chomsky (1965) the use of a distinct formal device for clausal recursion was abandoned, but the uniqueness of the sentence was retained.

Chomsky (1970), which introduced X’ theory, stated the ‘initial rule’ of the base to be  $S \rightarrow N'' V''$ , with no attempt to incorporate the sentence into the X’ system. (...)

Jackendoff (1977), trying to structurally generalize subject-of-sentence with possessor-of-NP (assuming the possessor is specifier of NP), proposes that sentence is the maximal projection of V. VP is a submaximal category, i.e. not a full phrase. (Auxiliaries are also specifiers.)

Gazdar, Pullum, & Sag (1982) have an analysis similar to Jackendoff’s (sentence=V’, ‘VP’=V’). The only difference is that they analyze auxiliaries as verbs. (...)

Sentence is  $I^{\max}$ . Chomsky (1981) mentions this as a possibility. (...)

A fim de propormos regras gramaticais do português afinadas com o modelo da teoria X-barras, iremos adotar as projeções IP e CP. Tentaremos formalizar propostas interessantes de descrição sintática da estrutura da frase em português (como Pontes, 1973; Lobato, 1975; Lemle, 1984; Luft, 1986; Perini, 1989, 2000; e Othero, 2006), **ampliando** seu poder de descrição em alguns casos, **atualizando** as análises em outros casos, utilizando, para esse fim, o modelo proposto pela teoria X-barras.

---

Chomsky (1986) popularized this analysis (...). This implementation has the advantage of providing a full X' structure".

### 3 O GRAMMAR PLAY E AS ANÁLISES DE OTHERO (2004)

*Only God can make a tree*<sup>46</sup>

Joyce Kilmer

Em Othero (2004), apresentamos uma análise da sentença simples do português e uma aplicação computacional dessa análise. Desenvolvemos um programa de *parsing* sintático, o *parser* Grammar Play. O Grammar Play analisa unicamente sentenças simples declarativas do PB, ou seja, sentenças que contenham apenas um verbo e que não sejam interrogativas<sup>47</sup>. Ele é capaz de dizer quais dessas sentenças são gramaticais e quais não são, além de atribuir às sentenças gramaticais uma estrutura sintagmática baseando-se no modelo da teoria X-barras tradicional. Além disso, a interface gráfica do Grammar Play permite que o usuário visualize a estrutura das sentenças tanto em formato de colchetes rotulados quanto em forma de um marcador sintagmático (ou árvore sintática). Abaixo temos o exemplo de uma sentença analisada pelo Grammar Play<sup>48</sup>:

---

<sup>46</sup> “Somente Deus pode fazer uma árvore”.

<sup>47</sup> Na verdade, ficaram de fora apenas as interrogativas-QU, do tipo *Quem é amigo da Maria? O que a Maria quer de presente?*, etc. Sentenças interrogativas do tipo sim-ou-não, que não alteram sua estrutura com relação à sua correspondente declarativa, não apresentam problemas. Compare as estruturas dos pares *O João é amigo da Maria. / O João é amigo da Maria?*; *A Maria quer chocolates de presente. / A Maria quer chocolates de presente?*.

<sup>48</sup> O Grammar Play está disponível para *download* em [http://www.geocities.com/gabriel\\_othero/](http://www.geocities.com/gabriel_othero/).

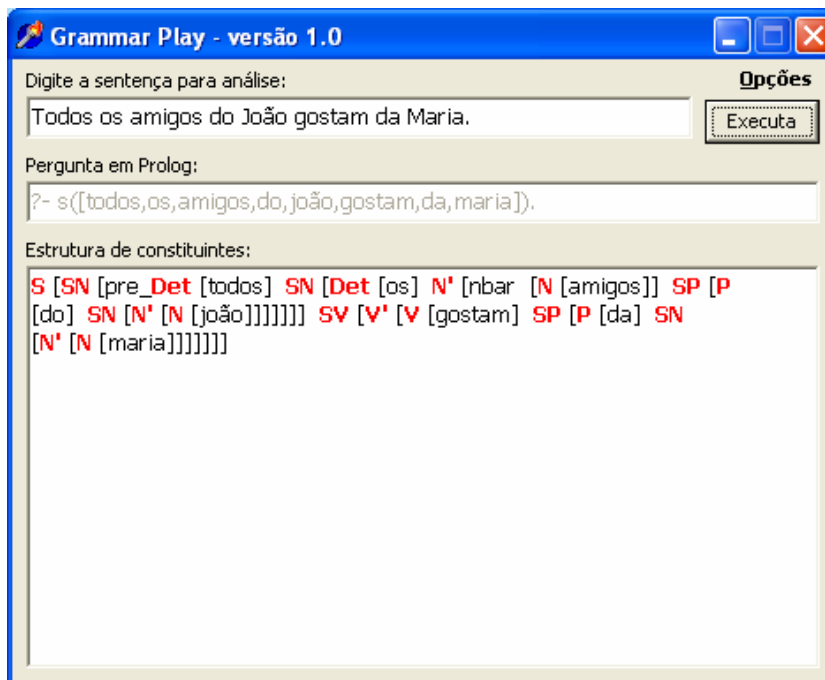


Figura 3.1: Exemplo de sentença analisada com colchetes rotulados

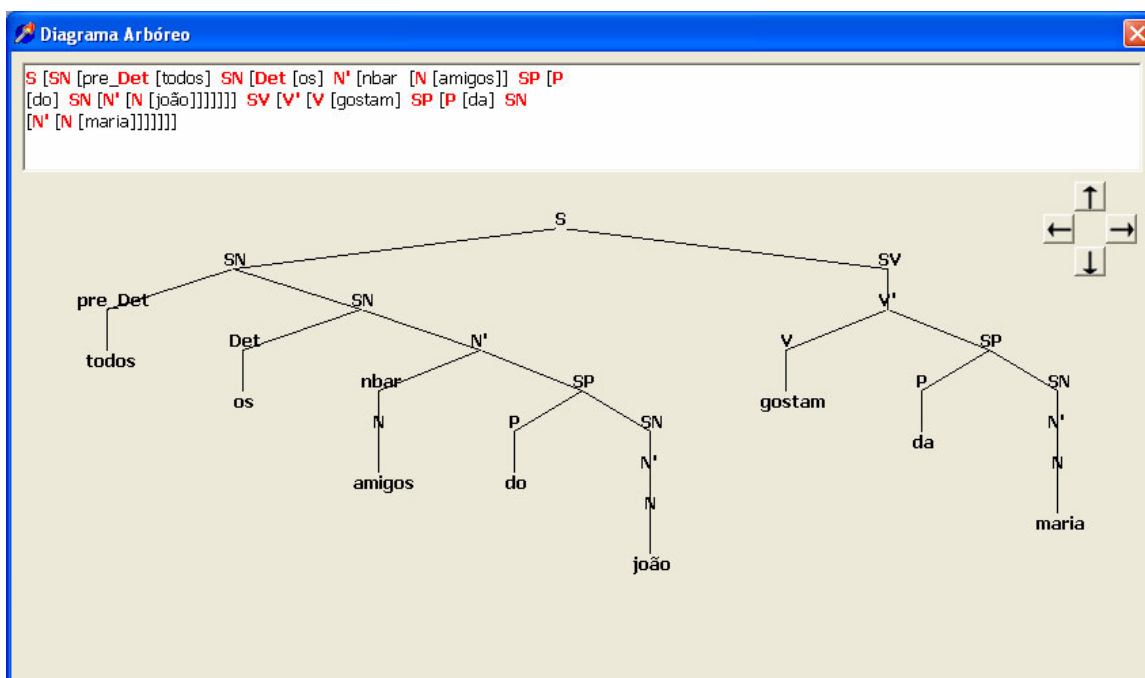


Figura 3.2: Exemplo de sentença analisada com sua estrutura arbórea

As duas ideias centrais subjacentes ao desenvolvimento do Grammar Play foram (i) a de apresentar um programa de *parsing* sintático para o PB que fosse *user-friendly* (desenvolvemos o programa pensando em todo aquele nosso colega linguista que não tem noções de programação, mas que deseja saber mais sobre a interface da Sintaxe com a

computação); e (ii) a de implementar computacionalmente uma descrição do português de acordo com uma teoria linguística consistente (em nosso caso, uma versão da teoria X-barra).

Para pegar um trecho emprestado de Menuzzi & Othero (2008: 1):

O parser [Grammar Play] é uma primeira tentativa de expandir a cobertura de analisadores semelhantes, como o esboçado em Pagani (2004) e Othero (2004). Os objetivos que guiam a presente versão do Grammar Play são o de implementar computacionalmente modelos lingüísticos coerentes aplicados à descrição do português e o de criar uma ferramenta computacional que possa ser usada didaticamente em aulas de introdução à sintaxe e lingüística, por exemplo.

As regras da gramática do *parser* foram elaboradas a partir de regras relativamente simples já consagradas na literatura pertinente sobre descrição sintática do português e a partir da análise de exemplos que foram encontrados em livros de introdução à teoria sintática. A gramática do Grammar Play desde o início seguiu o modelo de uma gramática PSG livre de contexto enriquecida com traços lexicais. Tentamos seguir os moldes do esquema apresentado pela teoria X-barra tradicional. Todas as regras foram adaptadas à linguagem Prolog, com o recurso das *Definite Clause Grammars* (DCGs)<sup>49</sup>.

As regras de descrição sintática que apresentamos foram as seguintes<sup>50</sup>:

- (1)  $S \rightarrow NP VP$
- (2)  $S \rightarrow VP NP$
- (3)  $S \rightarrow VP$
- (4)  $S \rightarrow AdvP S$
  
- (5)  $NP \rightarrow det N'$
- (6)  $NP \rightarrow N'$
- (7)  $NP \rightarrow pré-det NP$
- (8)  $N' \rightarrow Pro$ <sup>51</sup>
- (9)  $N' \rightarrow N$

<sup>49</sup> Julgamos não ser pertinente discutir o recurso das DCGs aqui. Sobre o assunto, remetemos o leitor a Pereira & Warren (1980), Pereira & Shieber (1987), Bratko (1997) e Brna (2001).

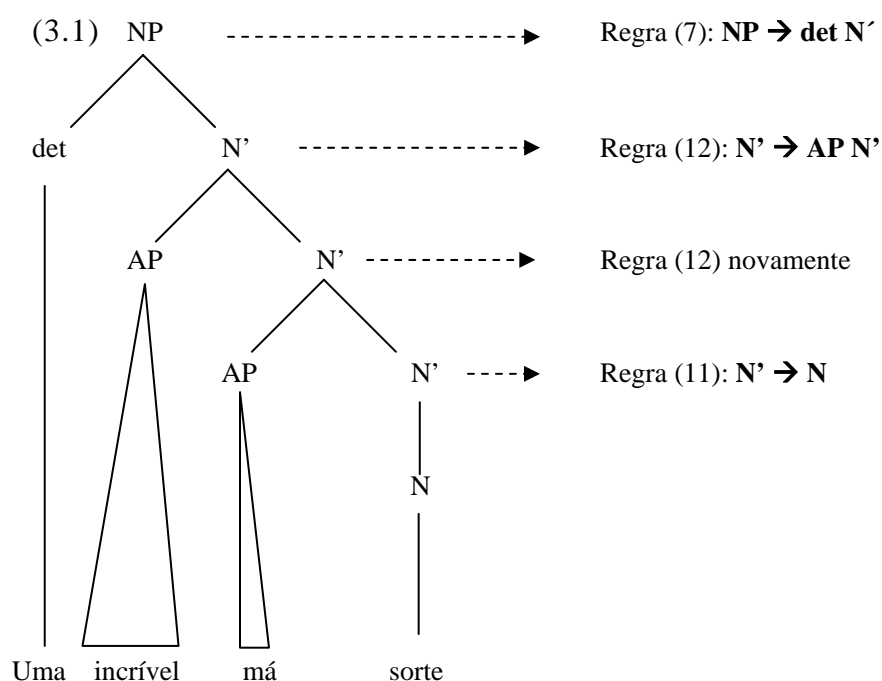
<sup>50</sup> Estamos apresentando essas regras porque partiremos delas para efetuarmos nossas análises no capítulo 4.

<sup>51</sup> “Pro” é um pronome. Não confundir com **pro** (“prozinho”) nem com **PRO** (“prozão”).

- (10)  $N' \rightarrow AP N'$
- (11)  $N' \rightarrow N' AP$
- (12)  $N' \rightarrow N' PP$
  
- (13)  $AP \rightarrow Adj' AdvP$
- (14)  $AP \rightarrow Adj' PP$
- (15)  $AP \rightarrow AdvP Adj'$
- (16)  $AP \rightarrow Adj'$
- (17)  $Adj' \rightarrow Adj$
- (18)  $Adj' \rightarrow AdvP Adj'$
- (19)  $Adj' \rightarrow Adj' AdvP$
- (20)  $Adj' \rightarrow Adj' PP$
  
- (21)  $PP \rightarrow P NP$
- (22)  $PP \rightarrow P AdvP$
  
- (23)  $VP \rightarrow V'$
- (24)  $VP \rightarrow V' PP$
- (25)  $VP \rightarrow AdvP V'$
- (26)  $VP \rightarrow V' AdvP$
- (27)  $V' \rightarrow V' NP$
- (28)  $V' \rightarrow V' PP$
- (29)  $V' \rightarrow AdvP V'$
- (30)  $V' \rightarrow V' AdvP$
- (31)  $V' \rightarrow V$
- (32)  $V' \rightarrow V NP$
- (33)  $V' \rightarrow V PP$
- (34)  $V' \rightarrow V AP$
- (35)  $V' \rightarrow V AdvP$

- (36) AdvP  $\rightarrow$  Adv'
- (37) AdvP  $\rightarrow$  AdvP Adv'
- (38) AdvP  $\rightarrow$  Adv' PP
- (39) Adv'  $\rightarrow$  Adv
- (40) Adv'  $\rightarrow$  Adv' PP

Essas 40 regras de descrição da sentença simples em PB resultaram em 67 regras implementadas na gramática do Grammar Play. Essa diferença no número total de regras se deve principalmente ao fator **hipergeração**. Ou seja, as regras de descrição sintática que elaboramos estão teoricamente corretas, mas elas hipergeram estruturas e hipergeneralizam as análises do *input* linguístico que o *parser* recebe. As regras (1) a (40) são o que chamamos de **macrorregras** sintáticas. E essa noção será importante para o desenvolvimento de nosso trabalho no próximo capítulo. Por exemplo, poderíamos analisar o NP [uma incrível má sorte], utilizando as regras (7), (12) e (11) – sem entrar em detalhes na estrutura do AP. Confira:



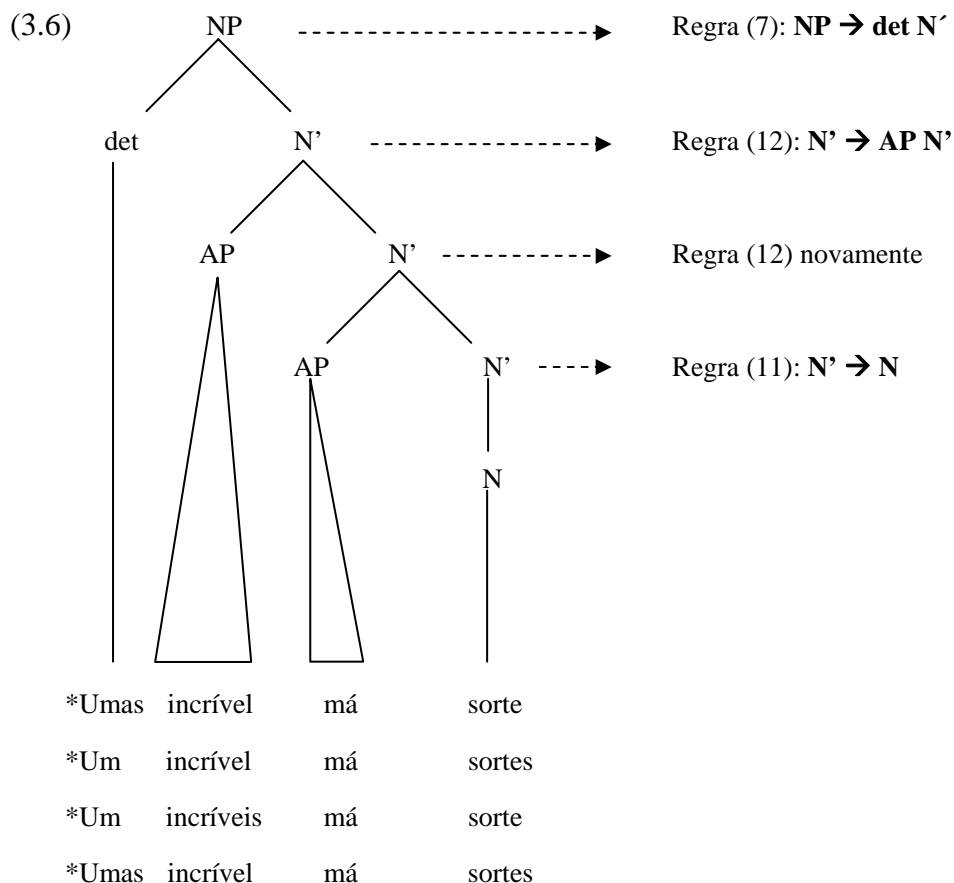
Porém, essas mesmas regras poderiam gerar\analisar os NPs agramaticais (3.2), (3.3), (3.4) e (3.5), como mostramos em (3.6):

(3.2) \*Umas incrível má sorte

(3.3) \*Um incrível má sortes

(3.4) \*Um incrível má sorte

(3.5) \*Umas incrível má sortes



O problema é que essas macrorregas sintáticas, tais como apresentamos de (1) a (40), não preveem a **unificação** (ou checagem) **de traços lexicais**, hipergerando estruturas que são agramaticais na língua. Isso foi algo com que tivemos de nos preocupar em nossa implementação computacional da gramática em Prolog. Daí as 40 regras de descrição sintática da sentença simples do português se transformarem em 67 regras implementadas computacionalmente em Prolog. Todo esse trabalho nos lembrou uma frase de Ray Jackendoff (2002: 167): “o problema é que a sintaxe bonitinha sozinha não tem tanto uso para a compreensão da máquina”<sup>52</sup>.

Na verdade, o problema já é bem conhecido. David (2007: 13), em seu trabalho de

<sup>52</sup> Trecho original: “The problem is that fancy syntax alone isn’t that much use for machine understanding”.

implementar computacionalmente (também em Prolog) um fragmento de gramática do DP, por exemplo, reparou que

alguns autores, como Raposo (1992), entendem que, para gerar um fragmento de gramática de estrutura sintagmática do português, faz-se necessário que se determinem regras capazes de gerar e reconhecer essas expressões nominais como bem formadas ou gramaticais. Assim é que Raposo (1992) apresenta regras de reescrita categoriais, as quais não conseguem assegurar que os traços gramaticais, que diferentes palavras trazem em sua estrutura, possam ser compatíveis com aqueles de outras palavras da mesma expressão.

Por isso, tivemos de enriquecer os itens lexicais do *parser* com traços gramaticais que percolavam, de certa forma, os nós dos constituintes, desde os itens terminais até o nó máximo da sentença. Assim, garantimos a análise de boa formação dos sintagmas e da sentença.

Por exemplo, os verbos contêm a informação de **número** e de **subcategorização sintática**; e os substantivos, adjetivos e determinantes, a informação de **número** e **gênero**<sup>53</sup>. Essas informações presentes nos itens lexicais percolam os nós até o nó máximo, efetuando uma **unificação de traços**. Abaixo, mostramos a entrada de alguns itens lexicais no léxico do Grammar Play, para dar uma ideia clara do que estamos falando:

---

<sup>53</sup> Evidentemente, o léxico pode conter outros tipos de informação, além desses traços. Outros dois trabalhos de implementação sintática em Prolog de fragmentos do PB, como Pagani (2004) e David (2007), trazem diferentes traços lexicais, como gênero, número, pessoa, definitude, etc.

<p>% Verbos Intransitivos (i) v(sing, i, morre). v(plur, i, morrem).</p> <p>% Verbos Transitivos Diretos (td) v(sing, td, come). v(plur, td, comem).</p> <p>% Verbos Transitivos Indiretos (ti) v(sing, ti(de), precisa). v(plur, ti(de), precisam).</p> <p>% Verbos Bitransitivos (tdi) v(sing, tdi(em), coloca). v(plur, tdi(em), colocam).</p> <p>% Verbos de Ligação (vl) v(sing, vl, continua). v(plur, vl, continuam).</p>
--

**Quadro 3.1:** Verbos no léxico do Grammar Play

<p>% Substantivos n([fem,sing], amiga). n([fem,plur], amigas). n([masc,sing], amigo). n([masc,plur], amigos).</p> <p>% Adjetivos adj([fem,sing], alta). adj([fem,plur], altas). adj([masc,sing], alto). adj([masc,plur], altos).</p>
--

**Quadro 3.2:** Substantivos e adjetivos no léxico do Grammar Play

Esses traços que aparecem entre parênteses nos itens lexicais estão presentes também nas regras de reescrita que implementamos na gramática do *parser*. Com uma descrição sintática “limpa” ou “pura”, nos moldes da teoria X-barra, como costumamos encontrar em manuais de descrição sintática do português<sup>54</sup>, acabamos formulando macrorreglas gramaticais.

Abaixo, apresentamos as regras sintagmáticas que desenvolvemos em nossa gramática em Prolog. Repare como as informações dos traços morfossintáticos presentes

<sup>54</sup> Como, por exemplo, Lemle (1984), Lobato (1986), Raposo (1992), Miotto et al. (2004), entre outros.

no léxico também aparecem nas regras categoriais de reescrita.

% Sentença (S)	
1)	s([sn, NP, sv, VP]) --> sn([_, Num], NP), sv(Num, VP).
2)	s([sv, VP, sn, NP]) --> sv(Num, VP), sn([_, Num], NP).
3)	s([sv, SVimp]) --> svimp(_, SVimp).
4)	S([sadv, AdvP, s, S]) --> sadv(AdvP), s(S).

**Quadro 3.3:** Regras da sentença

5)	sn(Conc, [det, [Det], n_bar, N_Barra]) --> det(Conc, Det), n_barra(Conc, N_Barra).
6)	sn(Conc, [n_bar, N_Barra]) --> n_barra(Conc, N_Barra).
7)	sn(Conc, [pre_det, [Pre_Det], sn, NP]) --> pre_det(Conc, Pre_Det), sn(Conc, NP).
8)	n_barra(Conc, [pro, [Pro]]) --> pro(Conc, Pro).
9)	n_barra(Conc, [n, [N]]) --> n(Conc, N).
10)	n_barra(Conc, [sadj, AP, n_bar, N_Barra]) --> sadj(Conc, AP), n_barra(Conc, N_Barra).
11)	n_barra(Conc, [n_bar, X, sadj, AP]) --> x(Conc, X), sadj(Conc, AP).
12)	x(Conc, [n, [N]]) --> n(Conc, N).
13)	x(Conc, [n_bar, Y, sadj, AP]) --> y(Conc, Y), sadj(Conc, AP).
14)	y(Conc, [n, [N]]) --> n(Conc, N).
15)	y(Conc, [n_bar, Z, sadj, AP]) --> z(Conc, Z), sadj(Conc, AP).
16)	z(Conc, [n, [N]]) --> n(Conc, N).
17)	n_barra(Conc, [n_bar, X, sp, PP]) --> x(Conc, X), sp(P, PP).

**Quadro 3.4:** Regras do NP

- |     |   |
|-----|---|
| 18) | sadj(Conc, [adj_bar, Adj_Barra, sadv, AdvP]) --> adj_barra(Conc, Adj_Barra), sadv(AdvP).      |
| 19) | sadj(Conc, [adj_bar, Adj_Barra, sp, PP]) --> adj_barra(Conc, Adj_Barra), sp(P, PP).           |
| 20) | sadj(Conc, [sadv, AdvP, adj_bar, Adj_Barra]) --> sadv(AdvP), adj_barra(Conc, Adj_Barra).      |
| 21) | sadj(Conc, [adj_bar, Adj_Barra]) --> adj_barra(Conc, Adj_Barra).                              |
| 22) | adj_barra(Conc, [adj, [Adj]]) --> adj(Conc, Adj).   |
| 23) | adj_barra(Conc, [sadv, AdvP, adj_bar, Adj_Barra]) --> sadv(AdvP), adj_barra(Conc, Adj_Barra). |
| 24) | adj_barra(Conc, [adj_bar, A, sadv, AdvP]) --> a(Conc, A), sadv(AdvP).                         |
| 25) | adj_barra(Conc, [adj_bar, A, sp, PP]) --> a(Conc, A), sp(P, PP).                              |
| 26) | A(Conc, [adj, [Adj]]) --> adj(Conc, Adj).   |

**Quadro 3.5:** Regras do AP

- |     |  |
|-----|--|
| 27) | sp(Lex, [p, [P], sn, NP]) --> p(Lex, inf, P), sn(Conc, NP).  |
| 28) | sp(Lex, [p, [P], sn, NP]) --> p(Lex, Conc, P), sn(Conc, NP). |

**Quadro 3.6:** Regras do PP

- 30) sv(Num, [v\_bar, V\_Barra, sadv, AdvP ]) --> v\_barra(Num, i, V\_Barra), sadv(AdvP).
- 31) sv(Num, [v\_bar, V\_Barra, sp, PP]) --> v\_barra(Num, i, V\_Barra), sp(, PP).
- 32) sv(Num, [v\_bar, V\_Barra]) --> v\_barra(Num, , V\_Barra).
- 33) svimp(Num, [v\_bar, V\_Barra\_i]) --> v\_barra\_i(Num, , V\_Barra\_i).
- 34) sv(Num, [sadv, AdvP, v\_bar, V\_Barra]) --> sadv(AdvP), v\_barra(Num, , V\_Barra).
- 35) svimp(Num, [sadv, AdvP, v\_bar, V\_Barra\_i]) --> sadv(AdvP), v\_barra\_i(Num, , V\_Barra\_i).
- 36) sv(Num, [v\_bar, V\_Barra, sadv, AdvP ]) --> v\_barra(Num, , V\_Barra), sadv(AdvP).
- 37) svimp(Num, [v\_bar, V\_Barra\_i, sadv, AdvP]) --> v\_barra\_i(Num, , V\_Barra\_i), sadv(AdvP).
- 38) sv(Num, [v\_bar, V\_Barra, sn, NP]) --> v\_barra(Num, , V\_Barra), sn(, NP).
- 39) sv(Num, [v\_bar, V\_Barra, sp, PP]) --> v\_barra(Num, , V\_Barra), sp(, PP).
- 40) svimp(Num, [v\_bar, V\_Barra\_i, sp, PP]) --> v\_barra\_i(Num, i, V\_Barra\_i), sp(, PP).
- 41) v\_barra(Num, , [sadv, AdvP, v\_bar, V\_Barra]) --> sadv(AdvP), v\_barra(Num, , V\_Barra).
- 42) v\_barra\_i(Num, , [sadv, AdvP, v\_bar, V\_Barra\_i]) --> sadv(AdvP), v\_barra\_i(Num, , V\_Barra\_i).
- 43) v\_barra(Num, i, [v, [V]]) --> v(Num, i, V).
- 44) v\_barra(Num, td, [v, [V], sn, NP]) --> v(Num, td, V), sn(, NP).
- 45) v\_barra(Num, ti, [v, [V], sp, PP]) --> v(Num, ti(P), V), sp(P, PP).
- 46) v\_barra(Num, vl, [vl, [V], sn, NP]) --> v(Num, vl, V), sn(Conc, NP).
- 47) v\_barra(Num, vl, [vl, [V], sadj, AP]) --> v(Num, vl, V), sadj(Conc, AP).
- 48) v\_barra(Num, vl, [v, [V], sp, PP]) --> v(Num, vl, V), sp(, PP).
- 49) v\_barra(Num, vl, [v, [V], sadv, AdvP]) --> v(Num, vl, V), sadv(AdvP).
- 50) v\_barra\_i(, i, [v, [Vimp]]) --> vimp(, i, Vimp).
- 51) v\_barra\_i(, td, [v, [Vimp], sn, NP]) --> vimp(, td, Vimp), sn(, NP).
- 52) v\_barra\_i(, , [v\_bar, T, sadv, AdvP]) --> t(, , T), sadv(AdvP).
- 53) v\_barra(Num, , [v\_bar, T, sadv, AdvP]) --> t(Num, , T), sadv(AdvP).
- 54) v\_barra(Num, , [v\_bar, T, sp, PP]) --> t(Num, , T), sp(, PP).
- 55) v\_barra(Num, , [v\_bar, T, sn, NP]) --> t(Num, , T), sn(, NP).
- 56) t(Num, , [v, [V]]) --> v(Num, , V).
- 57) t(Num, , [v, [V], sp, PP]) --> v(Num, , V), sp(, PP).
- 58) t(Num, , [v, [V], sn, NP]) --> v(Num, , V), sn(, NP).
- 59) t(Num, , [v\_bar, TT, sadv, AdvP]) --> tt(Num, , TT), sadv(AdvP).
- 60) t(, , [v, [Vimp], sn, NP]) --> vimp(, td, Vimp), sn(, NP).
- 61) tt(Num, , [v, [V]]) --> v(Num, , V).

Quadro 3.7: Regras do VP

61)	sadv([adv_bar, Adv_Barra]) --> adv_barra(Adv_Barra).
62)	sadv([sadv, B, adv_bar, Adv_Barra]) --> b(B), adv_barra(Adv_Barra).
63)	sadv([adv_bar, Adv_Barra, sp, PP]) --> adv_barra(Adv_Barra), sp(P, PP).
64)	adv_barra([adv, [Adv]]) --> adv(Adv).
65)	adv_barra([adv_bar, C, sp, PP]) --> c(C), sp(P, PP).
66)	b([adv_bar, Adv_Barra]) --> adv_barra(Adv_Barra).
67)	c([adv, [Adv]]) --> adv(Adv).

**Quadro 3.8:** Regras do AdvP

Não entraremos em detalhes sobre a implementação das regras sintáticas na gramática do *parser* porque não iremos propor a implementação computacional das regras que iremos apresentar aqui (cf. Othero, 2006, para detalhes da formulação e implementação computacional dessas regras gramaticais). A partir do próximo capítulo, iremos descrever os sintagmas do PB e algumas estruturas sentenciais com **a formulação de macrorregas de descrição sintática**. Em outras palavras, a gramática das estruturas sintagmáticas do PB que iremos propor não irá falar nada sobre o papel de traços lexicais. Sabemos que a descrição do léxico é fundamental para que possamos, por exemplo, implementar as regras sintáticas em ambiente computacional de maneira mais eficiente, como no exemplo de uma gramática em Prolog, que vimos acima. No entanto, iremos nos ater ao estudo das regras sintagmáticas de boa formação dos sintagmas e de algumas estruturas da sentença em PB e apenas daremos sugestões sobre os traços lexicais. As regras que apresentaremos a partir do próximo capítulo – por mais detalhadas que estejam e por mais que tenhamos pensado sobre elas – serão, ainda assim, macrorregas, que certamente estarão sujeitas à hipergeração de estruturas. No entanto, isso não é um problema, uma vez que, estando bem formadas as regras gramaticais, bastará o trabalho com os traços lexicais para que elas possam ser implementadas computacionalmente.

Ao não atrelar nossas regras à implementação computacional em uma determinada linguagem de programação (como o Prolog, por exemplo), pretendemos apresentar ao sintaticista e ao linguista computacional uma gramática da sentença do português que possa ser passível de implementação computacional em qualquer linguagem de computador apta ao trabalho em PLN. Essa gramática irá apresentar regras formais de reescrita dos sintagmas e da sentença em PB.

## 4 ANÁLISES DOS SINTAGMAS DO PB

*When I was a graduate student, N. Chomsky would argue that writing a rule does not constitute a solution to a problem, but merely a statement of it*<sup>55</sup>.

Luigi Burzio

Dividimos este capítulo em seis grandes seções, dedicadas às quatro projeções lexicais máximas e à estrutura da sentença em PB. Primeiramente, analisamos a estrutura do sintagma nominal (NP); depois, passamos à análise do sintagma adjetival (AP), do sintagma adverbial (AdvP), do sintagma preposicional (PP) e, finalmente, do sintagma verbal (VP). Por fim, apresentamos nossas análises da estrutura da sentença (IP e CP). A sexta e última seção traz um resumo de nossas regras gramaticais.

### 4.1 O SINTAGMA NOMINAL

Como já mencionamos na seção 2.3, adotaremos em nossas análises o núcleo funcional **D** e sua projeção máxima, o **DP**. Para iniciar o estudo do sintagma nominal, partimos das análises que apresentamos no capítulo 3. O primeiro passo foi verificar as regras da estrutura do NP que já havíamos proposto anteriormente e tentar “atualizá-las” à nova proposta. Por conveniência, repetimos aqui as regras do sintagma nominal que havíamos apresentado:

(4.1) NP → det N'

(4.2) NP → N'

(4.3) NP → pré-det NP

(4.4) N' → Pro

(4.5) N' → N

---

<sup>55</sup>“Quando eu era um aluno de pós-graduação, Noam Chomsky dizia que escrever uma regra não era a solução de um problema; era antes a mera formulação do problema”.

(4.6)  $N' \rightarrow AP N'$

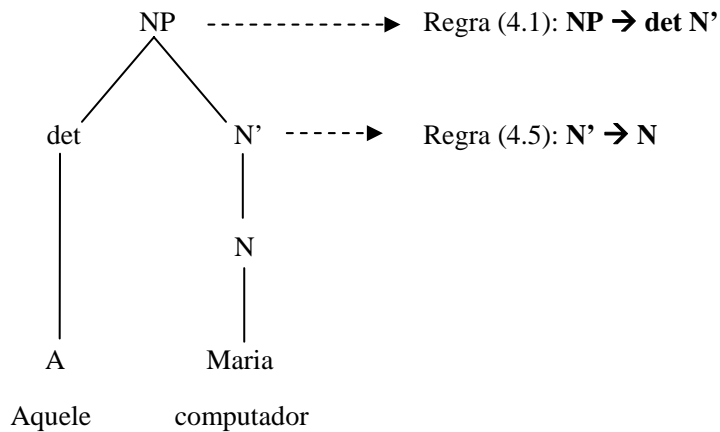
(4.7)  $N' \rightarrow N' AP$

(4.8)  $N' \rightarrow N' PP$

Essas oito regras gramaticais foram motivadas com base em análises de diversas sentenças e agrupamentos nominais, como os que apresentamos a seguir.

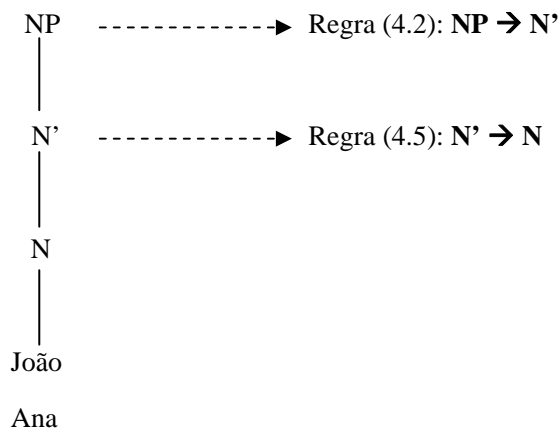
(4.9) [A Maria] adora chocolates.

(4.10) [Aquele computador] é melhor do que este.



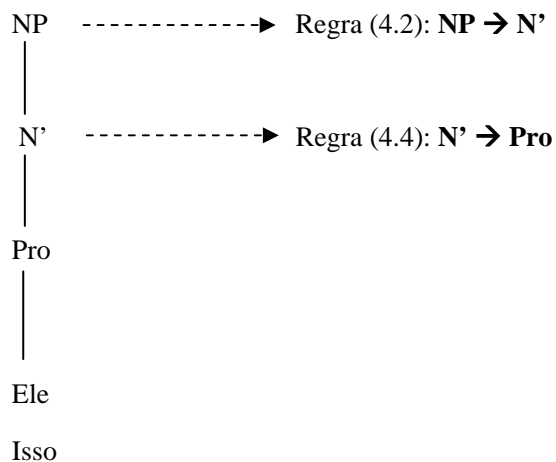
(4.11) [João] chegou.

(4.12) [Ana] saiu.



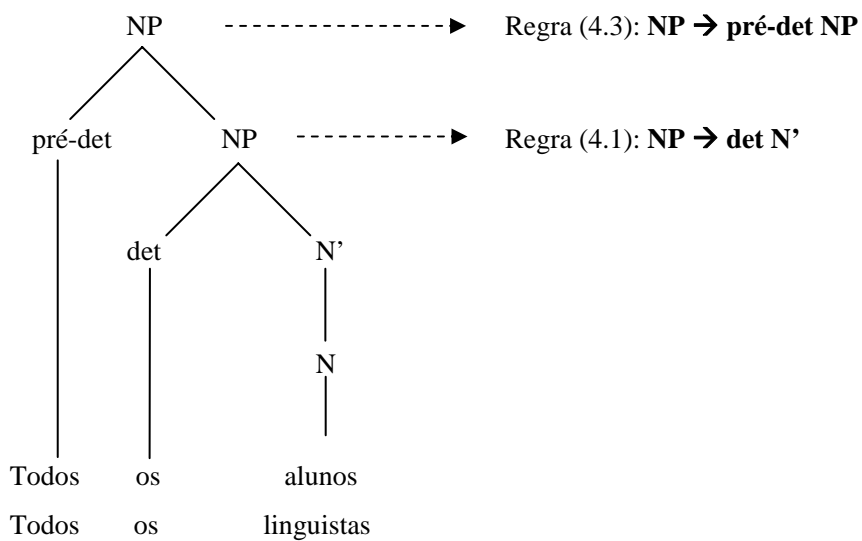
(4.13) [Ele] chegou.

(4.14) [Isso] é extremamente importante.



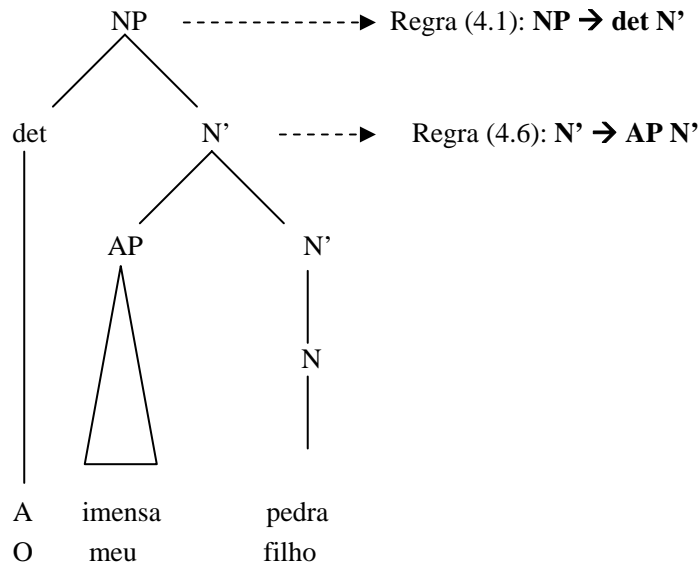
(4.15) [Todos os alunos] gostaram do livro.

(4.16) [Todos os linguistas] já leram algum texto do Chomsky.



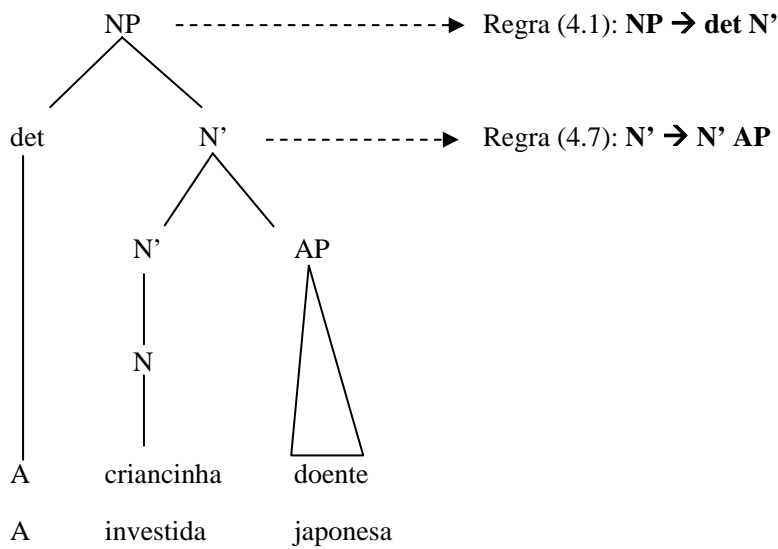
(4.17) [A imensa pedra] está bloqueando o caminho.

(4.18) [O meu filho] saiu.



(4.19) [A criancinha doente] melhorou.

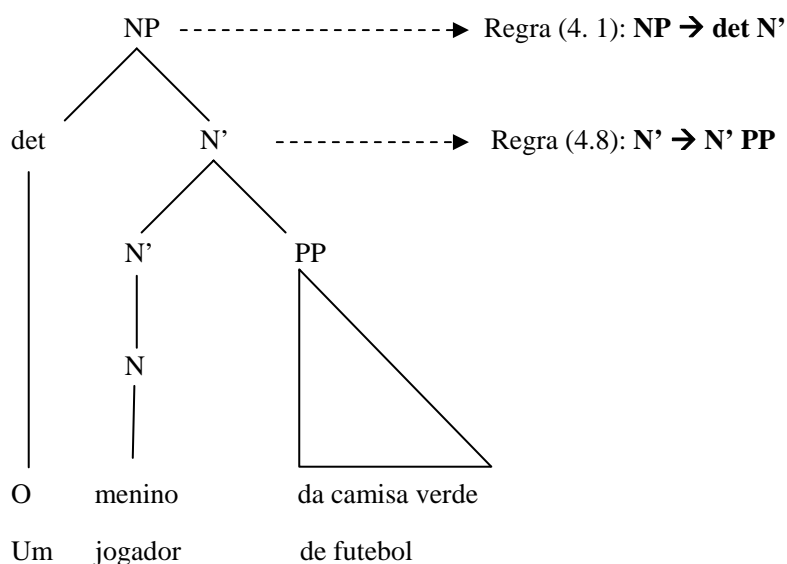
(4.20) [A investida japonesa] foi bem sucedida.



(4.21) [O menino da camisa verde] acaba de chegar.

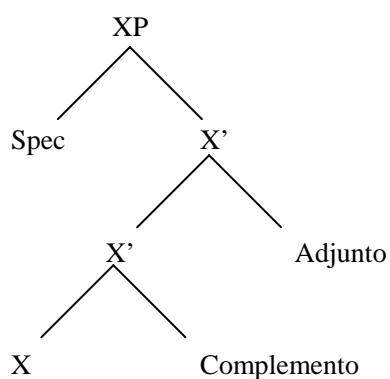
(4.22) [Um jogador de futebol] do Benfica<sup>56</sup>.

<sup>56</sup> Exemplo de Raposo (1992: 204). Voltaremos à discussão desse exemplo em seguida.



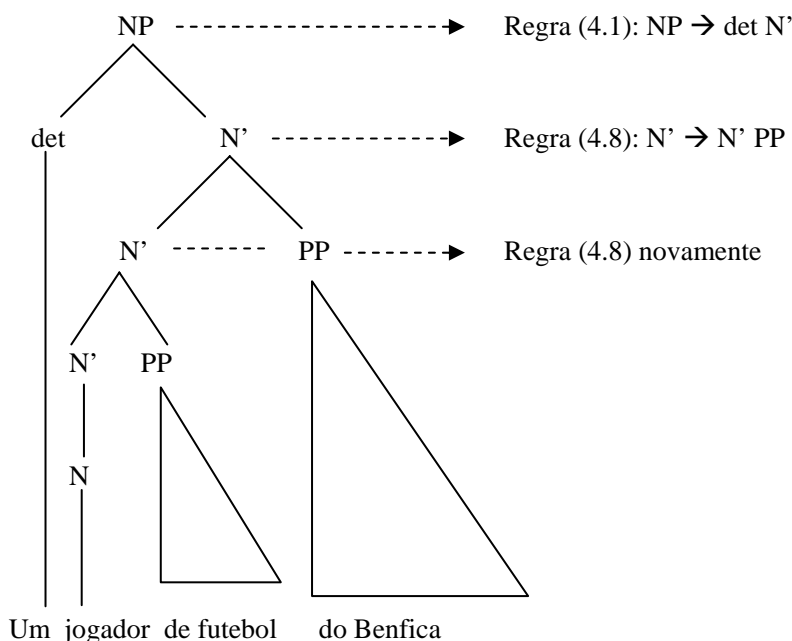
Essas regras nos pareceram interessantes à época e por isso resolvemos partir delas para seguir com nossas novas análises. Como vimos no capítulo 3, todas essas regras foram implementadas computacionalmente em Prolog na gramática do *parser* Grammar Play e se mostraram relativamente eficientes. Ou seja, elas estavam **formalizadas** o suficiente para que fossem implementadas em uma linguagem de programação para lidar com a sintaxe de uma língua natural. No entanto, além de não estarem afinadas com a nomenclatura corrente utilizada no modelo de descrição gramatical que adotamos, essas regras apresentam alguns problemas para a descrição dos sintagmas em português.

Por exemplo, um fator importante que o leitor pode ter percebido é que não há nenhuma regra entre (4.1) e (4.8) que preveja um complemento de N. No modelo de teoria X-barras que adotamos (cf. cap. 2), há uma diferença estrutural entre complementos e adjuntos que pode ser percebida nas relações de dominância estrutural. Reveja o esquema da X-barras abaixo:



De acordo com o modelo, o complemento está mais próximo do núcleo (é **irmão de X**), enquanto o adjunto vem ao lado do constituinte intermediário (é **irmão de X'**). Não havíamos feito essa distinção na estrutura interna do sintagma nominal nas regras que apresentamos acima. Por isso, as regras que envolvem APs e PPs mostram-nos sempre como adjuntos, ou seja, como irmãos da projeção intermediária do núcleo (**N'**), nunca como irmãos do nó terminal (**N**). Isso deve ter ficado claro especialmente na análise da sentença (4.22), de Raposo. Raposo (1992: 204) propõe que o sintagma preposicional [**de futebol**] seja complemento de N, enquanto [**do Benfica**] seja um adjunto. Contudo, de acordo com as regras que apresentamos acima, a análise completa do sintagma nominal em (4.22) não consegue dar conta dessa distinção:

(4.22) [Um jogador de futebol do Benfica]. (Raposo, 1992: 204)

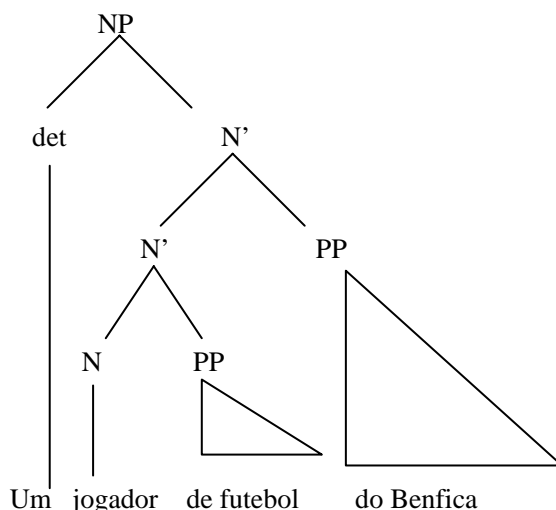


O que vemos acima é a repetição da regra (4.8), que diz que um **N'** pode expandir-se em um **N'** e um **PP**. Depois, temos a aplicação da regra (4.5), que autoriza que um **N'** possa expandir-se em um **N** e nada mais. Por isso, ambos os sintagmas preposicionais acima foram analisados como irmãos de **N'**, ou seja, como estando em posição típica de adjuntos. Essa é uma falha nas regras que apresentamos.

O mesmo acontece com os adjetivos e com os advérbios: eles são tratados como elementos **intransitivos**. Em outras palavras, em sua subcategorização, esses elementos

não recebem complementos. Essa análise, contudo, não parece ser a mais correta. Sobre o sintagma nominal de (4.22), Raposo (1992: 204) propõe a seguinte análise, por exemplo:

(4.22) [Um jogador de futebol do Benfica]. (Raposo, 1992: 204)



Na representação acima, o sintagma preposicional **[de futebol]** é irmão de N **[jogador]** (ou seja, é **complemento** de N), enquanto o sintagma preposicional **[do Benfica]** é irmão de N' **[jogador de futebol]** (ou seja, modifica N'), estando, portanto, em posição de **adjunto** do NP. Se essa análise for a mais correta, precisaremos reelaborar as regras de descrição do sintagma nominal em relação à posição dos complementos.

E como sabemos que a análise de Raposo sobre o status dos PPs **[de futebol]** e **[do Benfica]** está correta? Quais serão os critérios que utilizaremos para distinguir substantivos (e adjetivos e advérbios e preposições e verbos) que aceitam (ou exigem) complemento dos que não aceitam complemento?

Aqui faremos uma observação bastante importante: não entraremos no árduo assunto sobre a distinção entre **núcleos transitivos** e **núcleos intransitivos**. Há diversos trabalhos interessantes que discutem a distinção entre núcleos lexicais transitivos e intransitivos. O assunto tem sido estudado tanto por gramáticos tradicionais já clássicos (como Bechara, 1960; e Luft, 1986), como por sintaticistas formais pioneiros, (como Jackendoff, 1977; e Williams, 1975). Para uma introdução ao problema, remetemos o leitor a Rodrigues (2000) e Carnie (2002); para uma proposta interessante para o PB, remetemos o leitor a Perini (1989, 2000).

De nosso ponto de vista, de quem está elaborando regras gramaticais de reescrita

para o PB sem lidar com os **traços gramaticais** dos itens lexicais, a diferença entre os adjuntos e complementos do sintagma será meramente a sua posição estrutural na árvore. Em outras palavras, iremos elaborar **macrorregras gramaticais** que nos permitem analisar estruturas sintáticas cujo núcleo exige ou aceita complementos e regras gramaticais que nos possibilitem analisar sintagmas cujo núcleo não aceita ou não exige complementos. Não iremos, portanto, analisar os itens léxicos em si em suas propriedades, por vezes idiossincráticas, de transitividade e seleção categorial. Daí entendermos nossas regras gramaticais como macrorregras de descrição gramatical.

Uma implementação computacional das regras gramaticais que iremos propor, ou um trabalho sintático mais fino, deve, evidentemente, levar em consideração a distinção entre núcleos transitivos e intransitivos, se quiser evitar a hipergeração de estruturas. Esse tipo de trabalho deve adotar uma **abordagem lexicalista**. A ideia central da abordagem lexicalista é justamente restringir as regras sintáticas com traços presentes nos itens lexicais. Esses traços são expressos através de variáveis que devem ser unificadas, percolando os nós em uma árvore sintática para garantir a boa formação da sentença, como fizemos em Othero (2004, 2006), por exemplo, e como mostramos no capítulo anterior.

Quando mostrarmos um elemento como complemento ou como adjunto em nossas análises (como fizemos com os NPs do exemplo de Raposo, acima), estaremos seguindo nossa própria intuição como linguista e falante nativo do português, além de análises já consagradas em teoria sintática do português. O foco de nosso trabalho é elaborar regras gramaticais (ou macrorregras gramaticais) que **prevejam estruturas com complementos e estruturas com adjuntos**, pois é fato consensual que as projeções máximas lexicais podem apresentar elementos que sejam subcategorizados pelos núcleos e elementos “acessórios” que apenas modificam o núcleo, como adjuntos<sup>57</sup>.

Uma outra limitação nas regras que apresentamos anteriormente foi a inclusão de classes que não eram projeções máximas, como “det” e “pré-det”, uma saída reconhecidamente *ad hoc*, que vai contra o princípio de maximalidade (cf. cap. 2) da teoria X-barra. Essa crítica nos foi feita por Alencar (2008: 9):

Othero [2006] trata quantificadores em exemplos do tipo de (2) [todos, ambos] como pré-determinantes, inseridos por meio da regra NP → pré-det NP (p. 112). Essa análise viola um importante princípio do que seria uma teoria X-barra padrão: todo não-núcleo é um sintagma (Grewendorf,

---

<sup>57</sup> Sobre o assunto, cf. também Radford (1981, 1988) e Ernst (2001), além de Cançado (2003, 2005) para uma problematização sobre a distinção entre complementos e adjuntos de um ponto de vista semântico.

Hamm & Sternefeld 1989: 202). Uma análise em termos da hipótese DP que obedecesse a esse princípio trataria todas, por exemplo, como QP na posição de especificador do DP. Othero, em vez disso, trata esse quantificador como adjunto, de modo que o analisador gramatical hipergera. Com efeito, a frase agramatical (11) é reconhecida como gramatical pelo Grammar Play:

(11) \*Todas ambas ambas todas ambas as meninas dormem tranquilamente.

Para superar limitações como essas, reformulamos as regras para a descrição do sintagma nominal. Na verdade, o sintagma nominal como apresentamos nas regras (4.1) a (4.8) deu lugar às regras do DP, de (4.23) a (4.27):

(4.23)  $DP \rightarrow D'$

(4.24)  $D' \rightarrow D$

(4.25)  $D' \rightarrow D NP$

(4.26)  $D' \rightarrow D PossP$

(4.27)  $D' \rightarrow D NumP$

Além dessas regras, elaboramos as seguintes regras gramaticais para o **NP**, o **NumP**, o **PossP** e o **QP**:

(4.28)  $NP \rightarrow N'$

(4.29)  $N' \rightarrow N$

(4.30)  $N' \rightarrow N' AP$

(4.31)  $N' \rightarrow AP N'$

(4.32)  $N' \rightarrow N' PP$

(4.33)  $N' \rightarrow N PP$

(4.34)  $NumP \rightarrow Num'$

(4.35)  $Num' \rightarrow Num NP$

(4.36)  $Num' \rightarrow Num PP$

(4.37)  $PossP \rightarrow Poss'$

(4.38)  $Poss' \rightarrow Poss NP$

(4.39)  $Poss' \rightarrow NP Poss$

(4.40)  $Poss' \rightarrow Poss NumP$

(4.41)  $QP \rightarrow Q'$

(4.42)  $Q' \rightarrow Q DP$

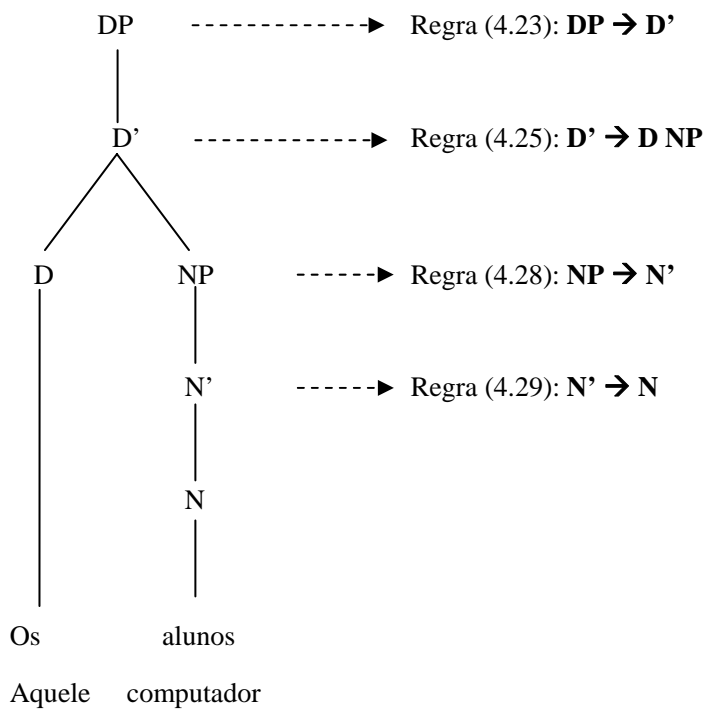
(4.43)  $Q' \rightarrow Q PP$

(4.44)  $Q' \rightarrow DP Q$

Vejamos a análise de alguns exemplos que motivaram nossas regras para discutir nossa proposta de descrição do DP.

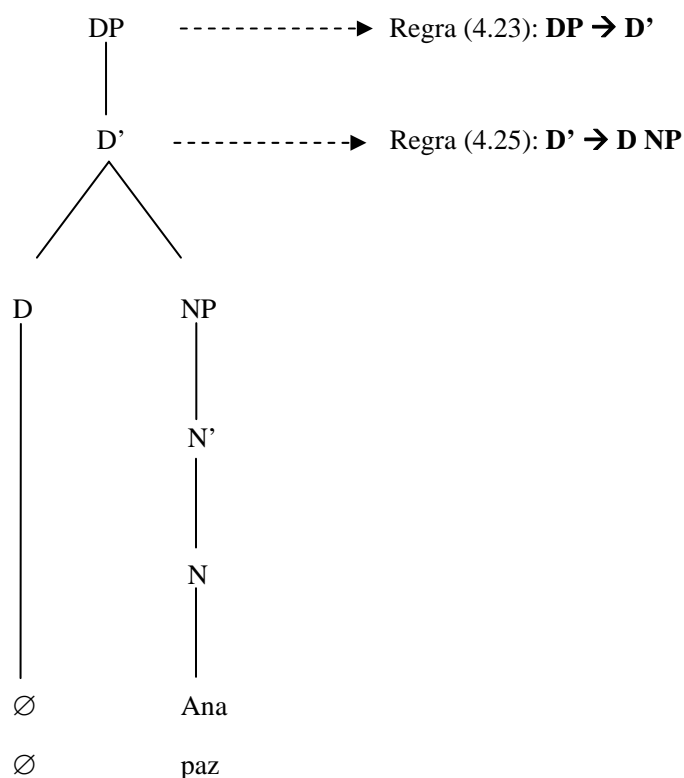
(4.45) [Os alunos] adoraram o livro.

(4.46) [Aquele computador] é melhor do que este.



(4.47) [Ana] chegou.

(4.48) Todos desejam [paz].



As estruturas arbóreas acima são talvez as mais consensuais na descrição do DP em português. A regra (4.25) permitiu a formação tanto dos DPs em (4.45) e (4.46) quanto dos DPs em (4.47) e (4.48), com o elemento D nulo (aqui representado pelo símbolo  $\emptyset$ ). Há fortes indícios de que o D possa ser expresso como elemento foneticamente não preenchido nas línguas<sup>58</sup>. Por isso, adotamos essa análise. Porém, sabemos que a implementação computacional de elementos vazios pode representar um problema<sup>59</sup>.

Vejamos a motivação para a regra (4.24).

(4.49) [Ele] comprou um livro interessante.

(4.50) [Isso] parece interessante.

<sup>58</sup> Remetemos o leitor a Giusti (1997), Vangsnes (2001), Pereltsvaig (2005) e algumas das referências citadas ali para discussão de alguns dos argumentos principais.

<sup>59</sup> No entanto, a proposta de Black (1997), que utiliza o formalismo PATR, consegue lidar com alguns tipos de movimento e vestígios na análise das sentenças em inglês. Cf. também algumas propostas de *parsing* em Prolog que adotam modelos da GB, como Kuhns (1986, 1990) e Chen, Lin & Wu (1988).



O núcleo do DP em português pode ser um **artigo**, um **elemento vazio**, um **pronome demonstrativo** ou um **pronome pessoal**. Os determinantes, assim como demais núcleos funcionais, subcategorizam o tipo sintático de seu complemento, ou seja, **selecionam** seus complementos. Por isso, encontramos dois tipos de núcleos D: aqueles que selecionam complementos (os complementos serão NPs, PossPs e NumPs, como veremos) e aqueles que são intransitivos, ou seja, que não selecionam complementos.

Em português, os itens lexicais que podem ocupar essa última posição, a de D intransitivo, são os **pronomes pessoais** e os **pronomes demonstrativos**, como mostramos nos exemplos (4.49) e (4.50). E os itens lexicais que podem ocupar a posição de D transitivos são os **artigos**, os **pronomes demonstrativos** (com exceção de **isso** e **aquilo**) e os **elementos-QU interrogativos**, como em **[Que livro] o João está lendo?**<sup>60</sup>.

Esses elementos estão em distribuição complementar na estrutura dos agrupamentos nominais em português e, por isso, são de mesma natureza. Não podemos ter as sequências abaixo, em português:

- (4.51) \*[O ele] chegou.
- (4.52) \*[Ele isso] chegou.
- (4.53) \*[A esta menina] saiu.
- (4.54) \*[Que este livro] João está lendo?
- (4.55) \*[Uma esta menina] gosta de Sintaxe.

<sup>60</sup> Alguns linguistas argumentam que os pronomes pessoais também podem ocupar a posição de núcleo D transitivos, como no caso dos DPs [*nós linguistas*], [*vocês meninas*], etc. Esses pronomes pessoais poderiam receber o traço [+transitivo]. Ignoraremos o fato aqui, pois não nos parece que o NP que segue o pronome esteja atuando como seu complemento. A esse respeito, cf. Radford (2007b), em análise do inglês, e David (2006a,b, 2007), em análise do português.

Etc.

Interessante que o mesmo não acontece na distribuição e no comportamento sintático de outros itens que costumamos encontrar nos agrupamentos nominais em PB, como os **pronomes possessivos** e os **numerais**, como vemos nas seguintes construções:

(4.56)

- a. [Meus sobrinhos]
- b. [Os meus sobrinhos]
- c. [Estes meus sobrinhos]
- d. [Estes meus dois sobrinhos]

(4.57)

- a. João engordou [dois porcos].
- b. João engordou [os dois porcos].
- c. João engordou [estes dois porcos].
- d. João engordou [seus dois porcos].
- e. João engordou [quais dois porcos]?

Os pronomes possessivos e os numerais costumam ser classificados como determinantes (ou como um tipo de determinantes) em PB<sup>61</sup>. No entanto, não podemos classificá-los como **determinantes**. Afinal, como vimos nos exemplos acima, eles não estão em distribuição complementar com os determinantes que havíamos estudado. E não podemos considerá-los adjetivos porque seu comportamento gramatical é distinto do comportamento dos adjetivos. Enquanto os adjetivos são núcleos lexicais, com conteúdo semântico (*bonito, rápido, grande*), os possessivos são núcleos funcionais que têm um papel gramatical dentro do agrupamento nominal<sup>62</sup>. Além do mais, os adjetivos podem ser utilizados de maneira recursiva para modificar um N (*querido amigo / querido prezado amigo / querido prezado excelentíssimo amigo*, etc.), enquanto os possessivos não têm

<sup>61</sup> Souza e Silva & Koch (1993), Bick (2000), Othero (2004, 2006). Há também quem considere os possessivos como uma subclasse dos adjetivos, por apresentarem comportamento sintático semelhante aos adjetivos na estrutura do DP, como Gonzaga (2003, 2006).

<sup>62</sup> Essa distinção “semântica” entre elemento **lexical** e elemento **funcional** remonta à tradição de gramáticos descritivos como Bolinger & Sears (1968), que utilizavam a dicotomia **palavras de conteúdo** (*contentives*) vs. **palavras de função** (*functors*). As primeiras têm conteúdo descritivo, semântico; as segundas trazem basicamente informações de caráter gramatical. Parece-nos justamente que os pronomes possessivos são “palavras de função”.

essa característica (*meu amigo* / *\*meu teu amigo* / *\*meu teu nosso amigo*, etc.). Uma outra evidência diz respeito à ordem de modificação dos adjetivos e dos possessivos dentro do DP (assunto que retomaremos com mais detalhes adiante, ainda neste capítulo): se os dois elementos encontram-se modificando um N em posição pré-nuclear, o adjetivo deve necessariamente estar mais próximo ao nome: *meu querido amigo* / *\*querido meu amigo*.

É comum encontrarmos os pronomes possessivos classificados como **D** em inglês (Radford, 1997a, b, por exemplo), já que, nessa língua, eles se encontram em distribuição complementar com os artigos e com os pronomes demonstrativos. Sua posição dentro do sintagma é intercambiável, e uma sequência com os dois elementos é agramatical, como mostramos abaixo:

(4.58) [My dog] is there.

(4.59) [The dog] is there.

(4.60) \*[The my dog] is there.

Em português, no entanto, todas essas sequências são gramaticais: *meu cachorro* / *o cachorro* / *o meu cachorro está lá*.

Uma outra saída comum adotada nos estudos da sintaxe do sintagma nominal em PB é classificar esses elementos como uma **subclasse dos determinantes**, ou seja, como pré- ou pós-determinantes. Foi a saída que adotamos em Othero (2004 e 2006), por exemplo, e que apresentamos no capítulo 3. Essa “solução” também não é boa. Ela apresenta, pelo menos, duas restrições graves, uma de ordem empírica e outra de natureza teórica.

Vejamos as seguintes sequências em português:

(4.61) [O computador]

(4.62) [O meu computador]

(4.63) [Os dois computadores]

O pronome possessivo aparece em posição pós-determinante em (4.62), assim como o numeral em (4.63). À primeira vista, poderíamos pensar em classificar esses dois elementos como pós-determinantes. No entanto, os dados abaixo nos mostram que esses elementos não são da mesma natureza, pois não estão em distribuição complementar:

(4.64) [Os meus dois computadores]

(4.65) \*[Os dois meus computadores]

(4.66) \*[Meus os dois computadores]

(4.67) \*[Dois os meus computadores]

(4.68) \*[Meus dois os computadores]

Os dois elementos podem co-ocorrer no agrupamento nominal, mas há uma ordem linear que deve ser respeitada entre eles. O pronome possessivo deve vir antes do numeral, e ambos devem figurar à direita do determinante. Isso parece complicar a classificação desses elementos dentro do NP.

Além disso, de um ponto de vista interno à teoria, se quisermos ser fiéis aos princípios teóricos que norteiam a descrição gramatical do modelo proposto pela teoria X-barra, não podemos postular elementos “soltos”, que não sejam projeções máximas, e que estejam desrespeitando, por exemplo, os princípios da lexicalidade e da maximalidade. Essa crítica, como já vimos, nos foi feita por Alencar (2008).

Para superar essas limitações, propusemos dois tipos de sintagma em PB que figuram dentro da estrutura do DP: o sintagma numeral (**NumP**) e o sintagma possessivo (**PossP**)<sup>63</sup>. Apresentaremos nossas análises de DPs em português que parecem mostrar que nossa proposta é adequada. Além de estudarmos com cuidado a estrutura interna dos agrupamentos nominais em PB para propor esses dois núcleos funcionais, nos beneficiamos da análise de Perini (2000), que também reconhece que os pronomes possessivos e os numerais não podem ser classificados como determinantes em português.

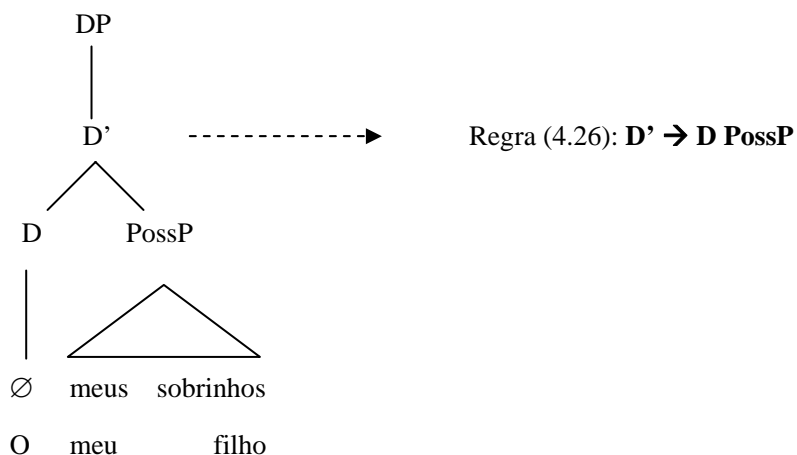
Vejamos algumas análises do DP com esses dois núcleos funcionais. Mais tarde, mostraremos a estrutura interna dos PossPs e dos NumPs.

(4.69) [Meus sobrinhos] acabam de sair.

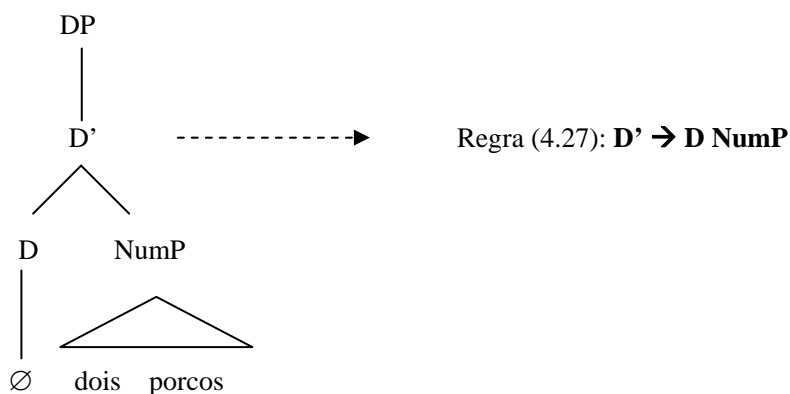
(4.70) [O meu filho] acaba de chegar.

---

<sup>63</sup> Sabemos que postular núcleos funcionais para descrever estruturas gramaticais na língua não é um recurso visto com bons olhos por alguns sintaticistas. Por isso, tivemos muito cuidado ao postular o NumP e o PossP. Realmente acreditamos que essas projeções podem esclarecer alguns pontos sobre a estrutura interna do DP em português. Recentemente (em agosto de 2008, durante o *VII Workshop on Formal Linguistics*, em Curitiba), ouvimos o seguinte comentário de um colega sintaticista: “não parece econômico ver que cada linguista postule um núcleo funcional na descrição da estrutura da sentença de uma dada língua. Mas, afinal de contas, quem garante que a estrutura gramatical de uma língua seja mesmo econômica? Se realmente houver 32 projeções dominando um sintagma verbal, temos de encarar isso como um fato da língua e não nos prender à economia que a teoria prega. Fazer o quê?”.



(4.71) João engordou [dois porcos].

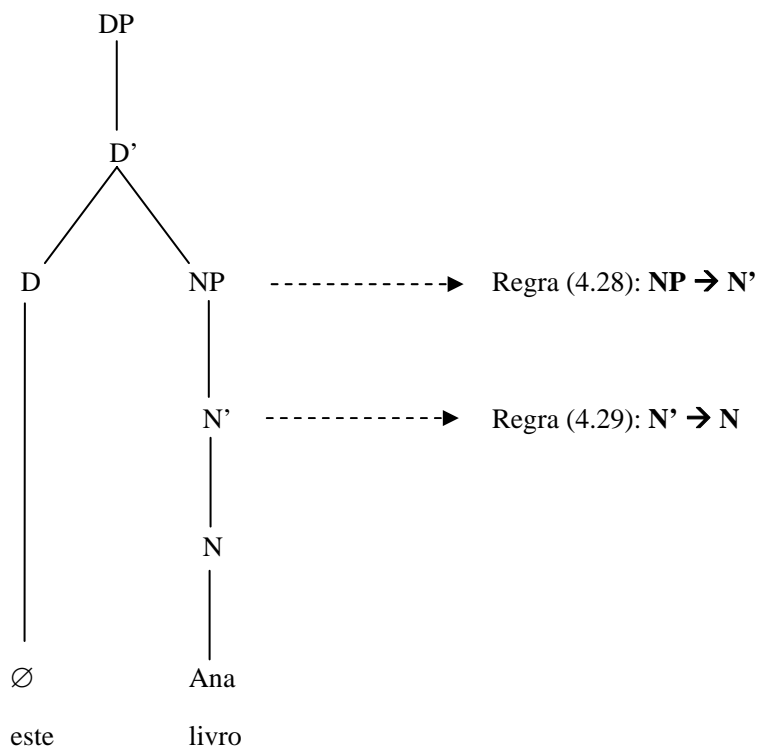


As regras do DP preveem estruturas com o núcleo D transitivo, que aceitam um complemento (NP, PossP ou NumP), e com um núcleo D intransitivo, que não necessita de complementos. Como dissemos, não estudaremos os traços lexicais que compõem cada elemento lexical. Mas podemos pensar em um traço  $[\pm\text{transitivo}]$  que poderia ser implementado na descrição do núcleo D. Os núcleos D transitivos c-selecionam como complemento um NP (regra 4.25), um PossP (regra 4.26) ou um NumP (regra 4.27).

Antes de analisarmos a estrutura interna do PossP e do NumP, passemos às regras do NP. Uma das limitações das regras que propusemos em Othero (2004), como vimos, foi a indistinção entre os complementos e adjuntos na estrutura interna do NP. Por isso, reelaboramos as regras para que elas pudessem mostrar essa diferença. Começamos com análises de NPs intransitivos, que não apresentam complementos, como os previstos pelas regras (4.29) a (4.32).

(4.72) [Ana] chegou.

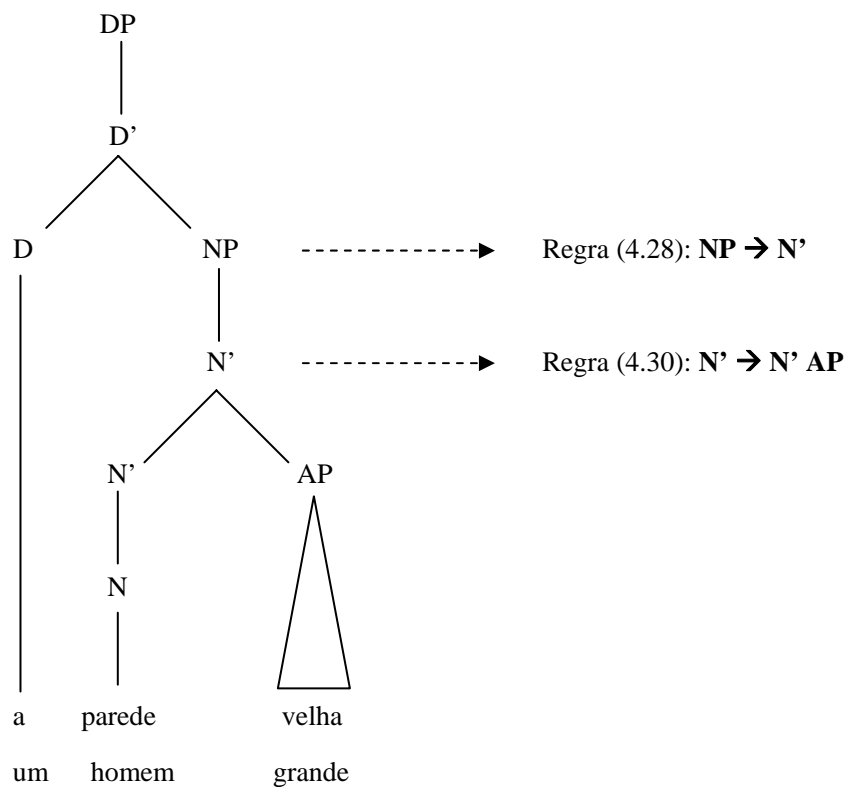
(4.73) [Este livro] parece interessante.



A estrutura arbórea acima apresenta um NP sem qualquer complemento ou adjunto. Mas sabemos que o NP pode apresentar complementos e adjuntos. Começemos por ver as regras que elaboramos para analisar NPs com adjuntos (os adjuntos do NP podem ser um **sintagma adjetival** (AP) ou **preposicional** (PP)).

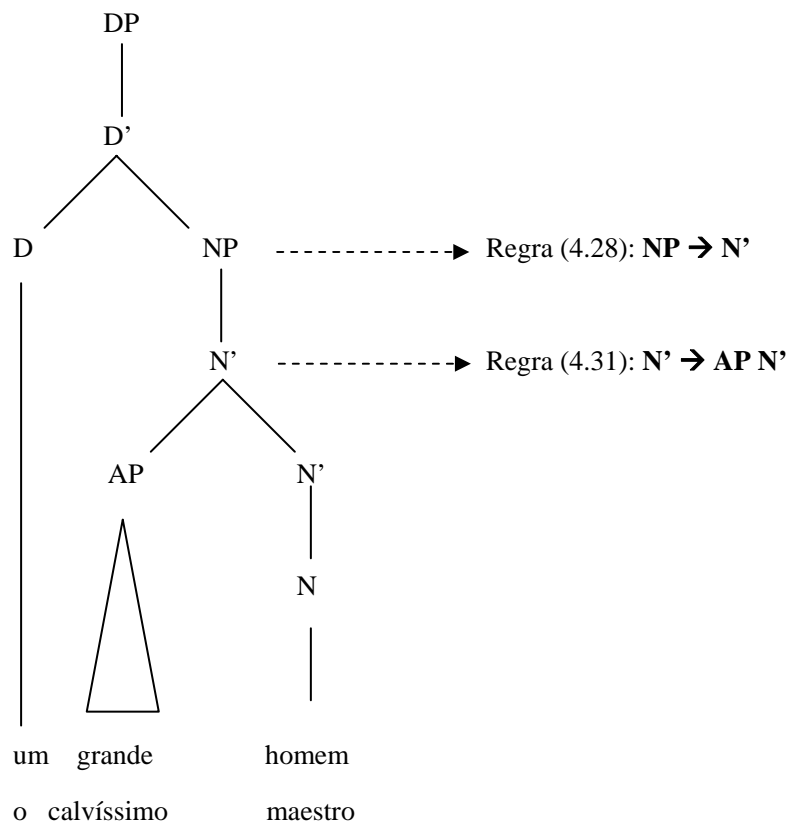
(4.74) Eu pintei [a parede velha].

(4.75) Oscar é [um homem grande].



(4.76) Oscar é [um grande homem].

(4.77) [O calvíssimo maestro] acaba de chegar.

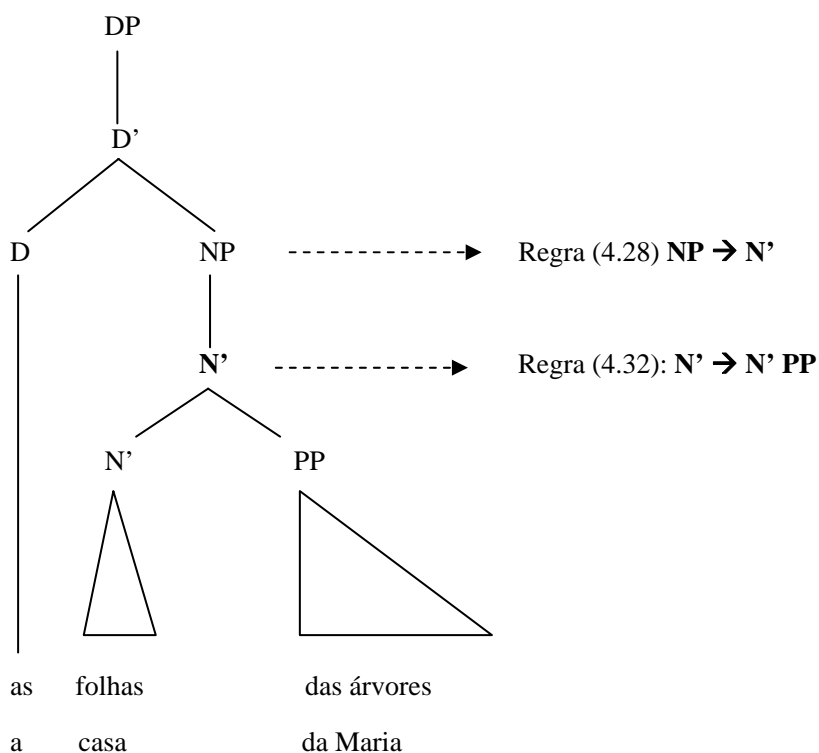


Em termos estritamente sintáticos e sintaticamente estritos, há pouca diferença entre as regras (4.30) e (4.31). A regra (4.30) prevê um AP como adjunto **à direita** do N', enquanto a regra (4.31) prevê um AP adjunto **à esquerda** do N'. No entanto, a modificação adjetival à esquerda ou à direita do N' envolve alguns aspectos semânticos interessantes (*um grande homem* não recebe a mesma interpretação que *um homem grande*, por exemplo)<sup>64</sup>.

Ainda tratando da modificação nominal, vemos a regra (4.32), que nos permite analisar NPs com modificadores preposicionais em posição de adjuntos.

(4.78) [As folhas das árvores] começaram a cair.

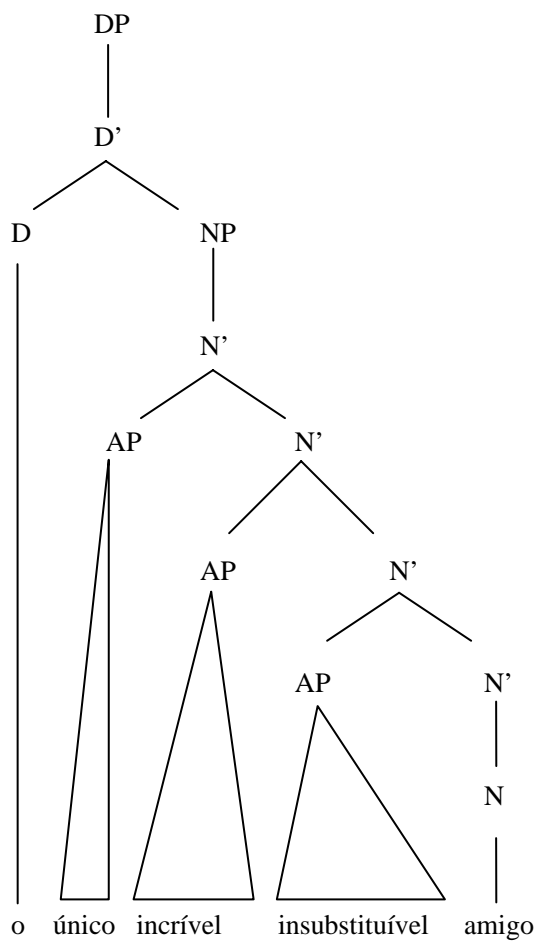
(4.79) João entrou [na casa da Maria].



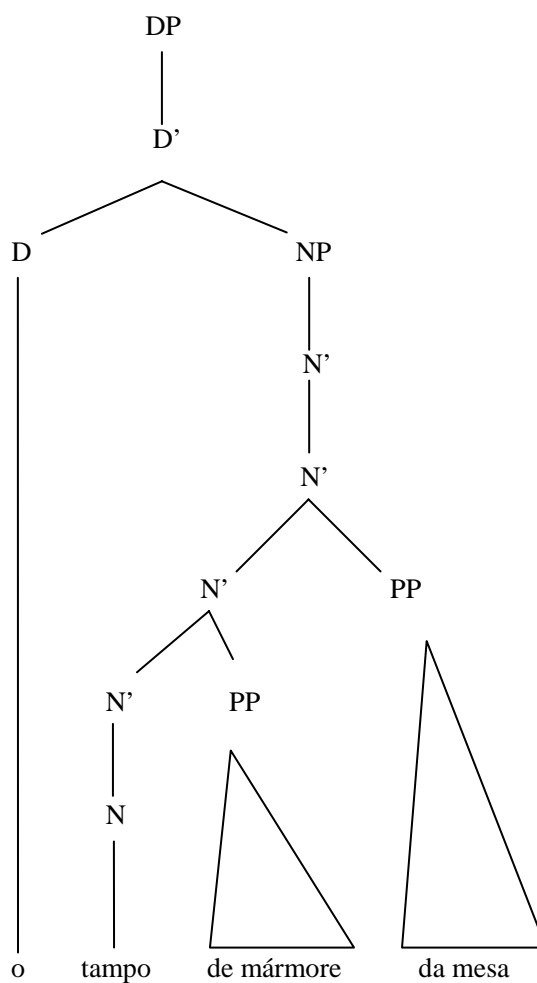
<sup>64</sup> Não lidaremos com o assunto aqui, já que essa distinção semântica não se reflete na estrutura sintática que estamos analisando além da distinção de ordem linear que já apontamos. Sobre o tema, cf. Bernstein (1991), Giorgi & Longobardi (1991), Valois (1991), além de dois trabalhos interessantes que lidam com dados do português: Menuzzi (1992), cuja proposta é descrever o comportamento e a interpretação de adjetivos pré- e pós-nominais em PB dentro da teoria gerativa; e Callou et al. (2003), que estudam a distribuição dos adjetivos dentro do NP em PB e PE, com base em corpora das duas línguas – os autores chegam à conclusão de que “no estudo da posição do Adjectivo no Sintagma Nominal intervêm factores muito diversos: morfossintáticos e sintáticos, de compatibilidade semântica e pragmática entre os adjectivos e os nomes, factores estilísticos e rítmicos, entre outros”.

Repare que essas regras são recursivas. Isso quer dizer que elas permitem a repetição de APs e PPs adjuntos ao N', como nas sentenças seguintes:

(4.80) [O único incrível insubstituível amigo] acaba de chegar.

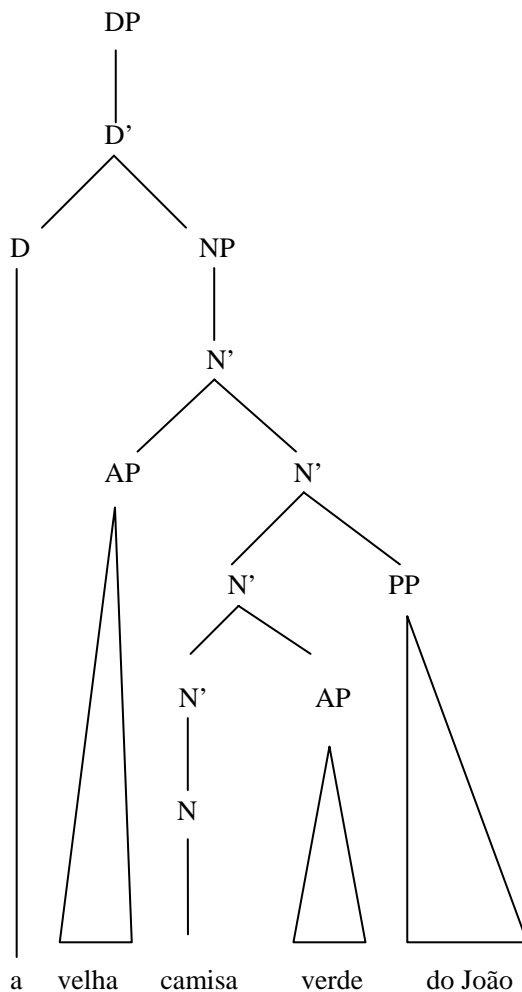


(4.81) [O tampo de mármore da mesa] rachou.



Os exemplos acima apresentam mais de um sintagma (adjetival ou preposicional) modificando o NP, o que é previsível pelas regras de nossa gramática. Além disso, as regras, como estão até agora, não permitem a formação de sintagmas agramaticais, com modificação de um PP à esquerda do N', como as sequências *\*o de mármore tampo* ou *\*as das árvores folhas*. E a modificação múltipla de um N' por APs e PPs também é prevista, como vemos na análise abaixo:

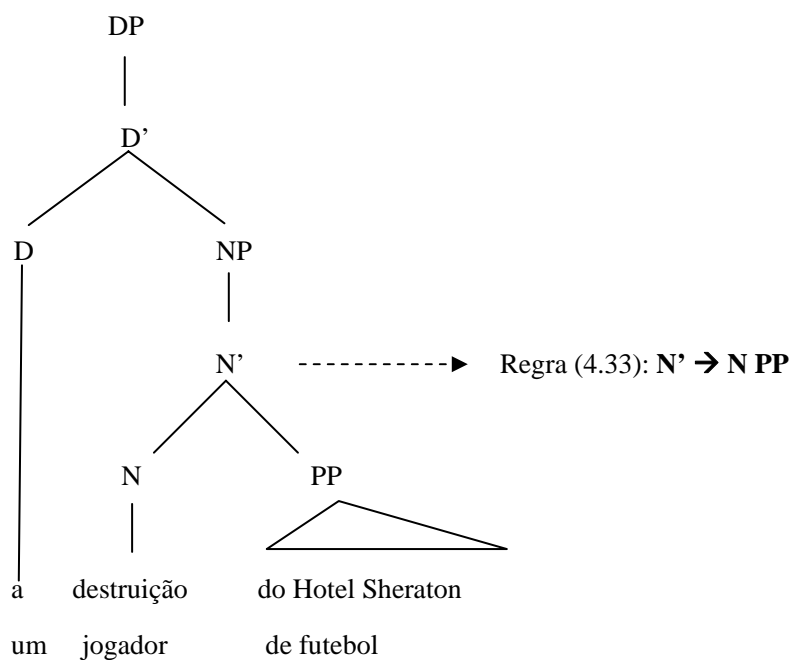
(4.82) [A velha camisa verde do João].



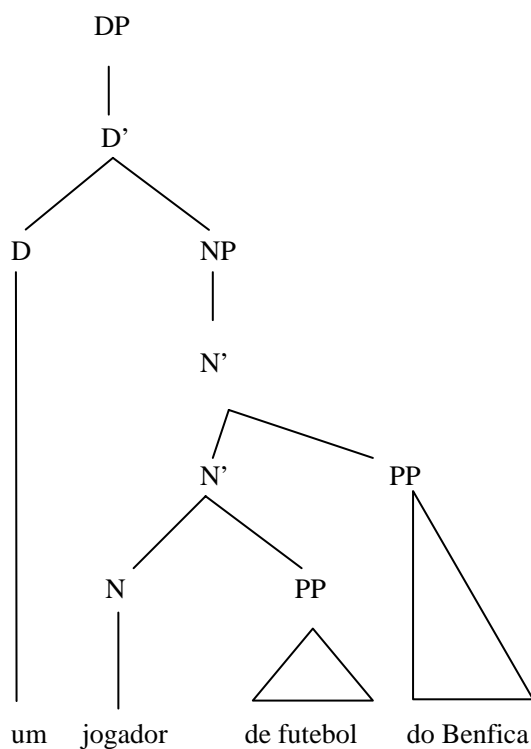
Essas análises demonstram que as regras gramaticais que formulamos para a descrição do DP e do NP em português parecem estar adequadas, prevendo a boa formação de sintagmas gramaticais e barrando alguns sintagmas agramaticais em PB. Passemos, por fim, à análise da regra (4.33), que prevê um N transitivo com complemento preposicional.

(4.83) Eu presenciei [a destruição do Hotel Sheraton].

(4.22) [Um jogador de futebol] do Benfica. (Raposo, 1992: 204)



Analisando o DP completo [**Um jogador de futebol do Benfica**], percebemos a distinção que as regras fazem entre complementos (irmãos do núcleo) e adjuntos (irmãos da projeção intermediária X')



Na estrutura arbórea acima, o PP [**de futebol**] é complemento do N – daí estar em

relação de irmandade com N –, enquanto o PP [**do Benfica**] é adjunto do NP – daí estar em relação de irmandade com o N' e não com o N.

Passemos às regras gramaticais (4.34) a (4.44), que tratam da descrição das projeções máximas funcionais que postulamos para a descrição do DP em português. Repetimos as regras aqui por conveniência:

(4.34) NumP → Num'

(4.35) Num' → Num NP

(4.36) Num' → Num PP

(4.37) PossP → Poss'

(4.38) Poss' → Poss NP

(4.39) Poss' → NP Poss

(4.40) Poss' → Poss NumP

(4.41) QP → Q'

(4.42) Q' → Q DP

(4.43) Q' → Q PP

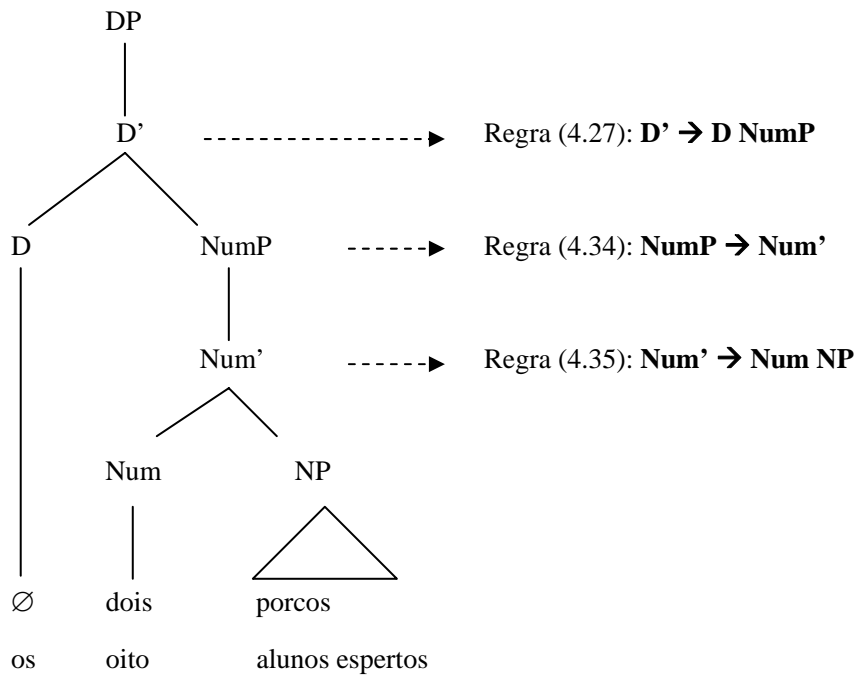
(4.44) Q' → DP Q

Postulamos a existência dessas categorias na estrutura do DP para dar conta de determinados agrupamentos nominais de maneira mais elegante e eficiente do que havíamos proposto em trabalhos anteriores. **Elegante** porque essas regras estão mais afinadas com a teoria gramatical da X-barras; **eficientes** porque acreditamos que, com essas regras, conseguimos analisar corretamente a estrutura interna do DP, lidando com o problema da ordenação de elementos como determinantes, pronomes possessivos e numerais no DP e NP.

Vejamos a motivação para as regras que apresentamos acima. Começemos pelo núcleo **Num**, expresso por um numeral. Vejamos a análise dos seguintes DPs:

(4.84) João engordou [dois porcos].

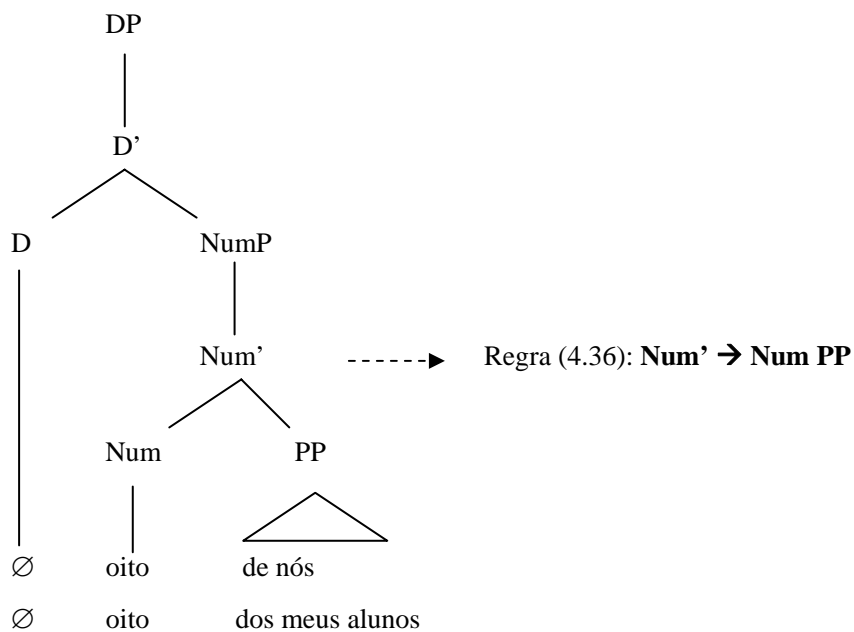
(4.85) [Os oito alunos espertos].



Repare que o **NumP** domina o **NP**. Graças a essa relação de dominância, conseguimos analisar com sucesso sintagmas como os seguintes:

(4.86) [Oito de nós].

(4.87) [Oito dos meus alunos].



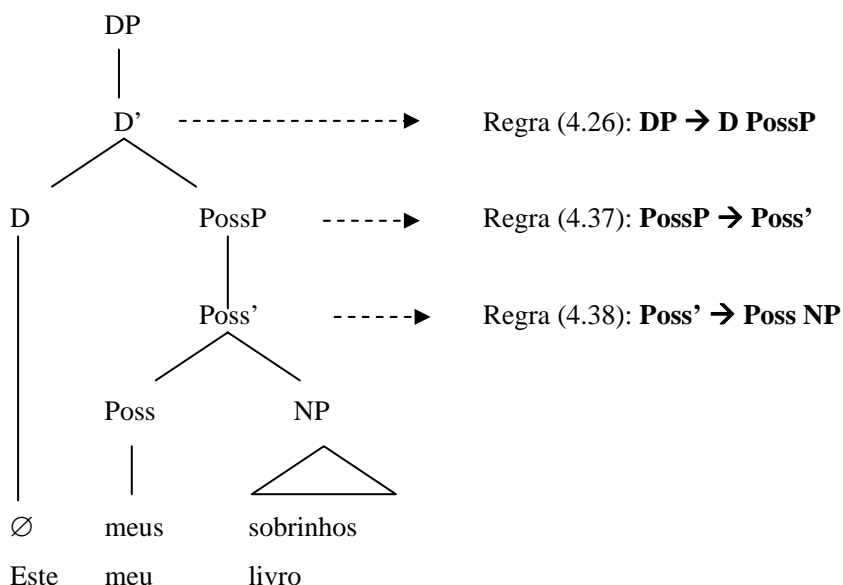
Uma vez que o sintagma máximo **NumP** contém o NP (e não está contido por ele),

podemos também estipular que NumP domine um PP, como mostramos na análise acima, em (4.86) e (4.87). De maneira correspondente, postulando que o DP domina o NumP, podemos também analisar sem problemas os sintagmas de (4.84) e (4.85), que apresentam um determinante foneticamente preenchido e um determinante vazio<sup>65</sup>. Além disso, essa análise será crucial em casos de sintagmas com pronomes possessivos: as regras do NumP irão interagir de maneira bastante harmônica com as regras referentes à estrutura do PossP, como veremos a seguir. Finalmente, uma outra vantagem dessa análise é que não é necessário postular que sequências como [o aluno], [meu aluno], [oito alunos], [alunos], etc. sejam de natureza diferente: todos são DPs e, portanto, sendo da mesma categoria, todos devem apresentar o mesmo comportamento sintático.

Passemos à análise de DPs contendo elementos possessivos.

(4.88) [Meus sobrinhos] são umas pestes.

(4.89) [Este meu livro] é muito interessante.



O DP deve dominar os núcleos funcionais NumP e PossP. Aqui, com o PossP, vemos mais um argumento para isso: se aceitarmos uma estrutura  $_{\text{PossP}}[ \text{DP}[ \dots ] ]$ , ou seja, com um PossP dominando um DP, estaríamos autorizando formações claramente agramaticais na língua, como *\*meus os sobrinhos*, *\*este meu este livro* e mesmo *\*meus*

<sup>65</sup> A ordem dos elementos do DP em português reflete a ordem hierárquica universal dos elementos dentro do agrupamento nominal proposta por Cinque (2005): **D** > **Num** > **Adj** > **N** (ainda que, como vimos, a ordem do adjetivo seja relativamente menos rígida, cf. Menuzzi, 1992, e Perini, 2000).

*meus sobrinhos* ou *\*este meu meu livro*. As regras acima preveem a agramaticalidade dessas sequências corretamente. Ao autorizarmos somente estruturas do tipo DP[ PossP[ NP[ ... ] ] ], evitamos esse tipo de má formação do sintagma.

Contudo, ainda não conseguimos lidar adequadamente com os DPs abaixo:

(4.90) A Maria leu [o meu livro].

(4.91) ?? A Maria leu [o livro meu]<sup>66</sup>.

(4.92) ?? A Maria leu [um meu livro].

(4.93) A Maria leu [um livro meu].

O que acontece aqui é que os pronomes possessivos se comportam de maneira distinta quando na presença de artigos definidos, como em (4.90) e (4.91), ou artigos indefinidos, como em (4.92) e (4.93). Isso sugere que o núcleo D deve conter um traço [ $\pm$  definido] que afetará a posição do pronome possessivo dentro do DP<sup>67</sup>. Em relação à sua posição no DP, encontramos o seguinte:

- com o artigo definido, o pronome possessivo deve ocupar posição prénominal (ver, no entanto, observação da nota 66);
- com o artigo indefinido, o pronome possessivo deve ocupar posição pós-nominal.

E quando temos um pronome demonstrativo, o possessivo pode ocupar qualquer uma das duas posições:

(4.94) A Maria leu [este meu livro].

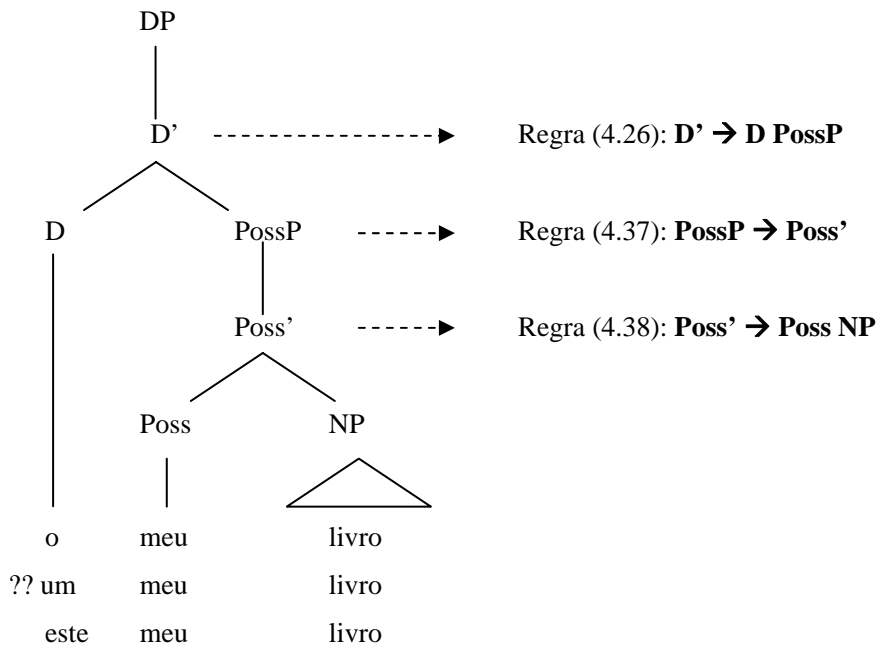
(4.95) A Maria leu [este livro meu].

---

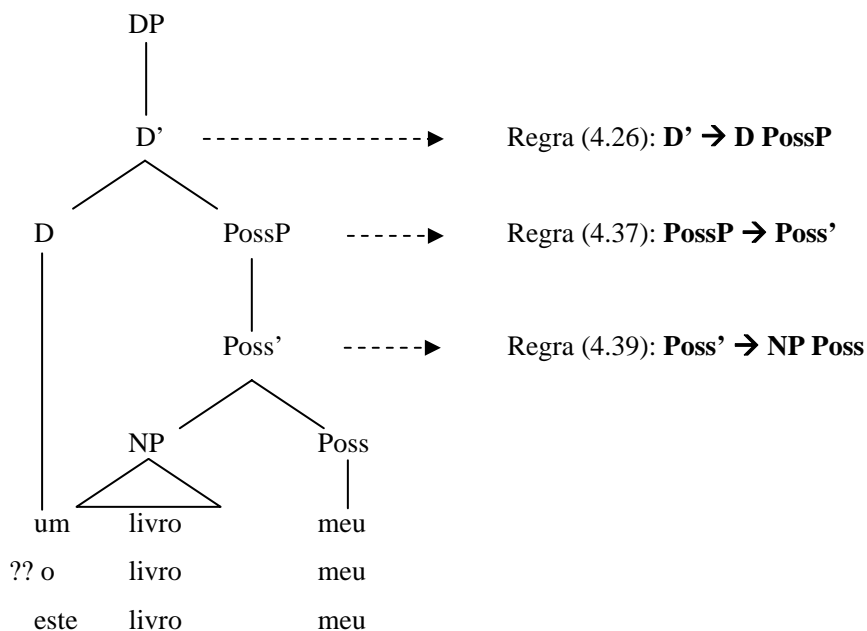
<sup>66</sup> Magdiel M. Aragão Neto, em comunicação pessoal, não julga essa sentença estranha, uma vez que, “na poesia, por exemplo, é muito comum essa estrutura e ninguém estranha”. Além do mais, encontramos esse tipo de ordem em PB, ainda que de maneira restrita, como neste trecho da oração do Pai Nosso: “**O pão nosso** de cada dia nos dai hoje (...)”. Essa estrutura era comum em português arcaico (Mattos e Silva, 2006) e ainda encontramos algumas ocorrências no português falado no Brasil (Moura Neves, 1993). Remetemos o leitor a Castro (2006a, b) e Brito (2007), que comparam o comportamento desse tipo de construção em PE e PB e trazem alternativas de análise para o fenômeno, nas duas variedades do português.

<sup>67</sup> A existência de um traço [ $\pm$  definido] é sugerida por Martinho (1998) e David (2007), para o português europeu e brasileiro, respectivamente. A mesma intuição também é compartilhada por Gonzaga (2003) ao lidar com o problema.

Isso não representa de fato um problema para nossas (**macro**)regras gramaticais, uma vez que nossas regras permitem a formação de estruturas bem formadas, como mostramos nas representações arbóreas abaixo. O problema, na verdade, é que não conseguimos restringir a formação de estruturas mal formadas na língua. Ou seja, a mesma regra que garante e prevê a boa formação do DP em (4.90) e (4.94) também garante a boa formação do estranho DP em (4.92).



O mesmo ocorre quando da descrição dos DPs (4.93) e (4.95): as regras preveem sua boa formação, mas não excluem o DP estranho de (4.92):

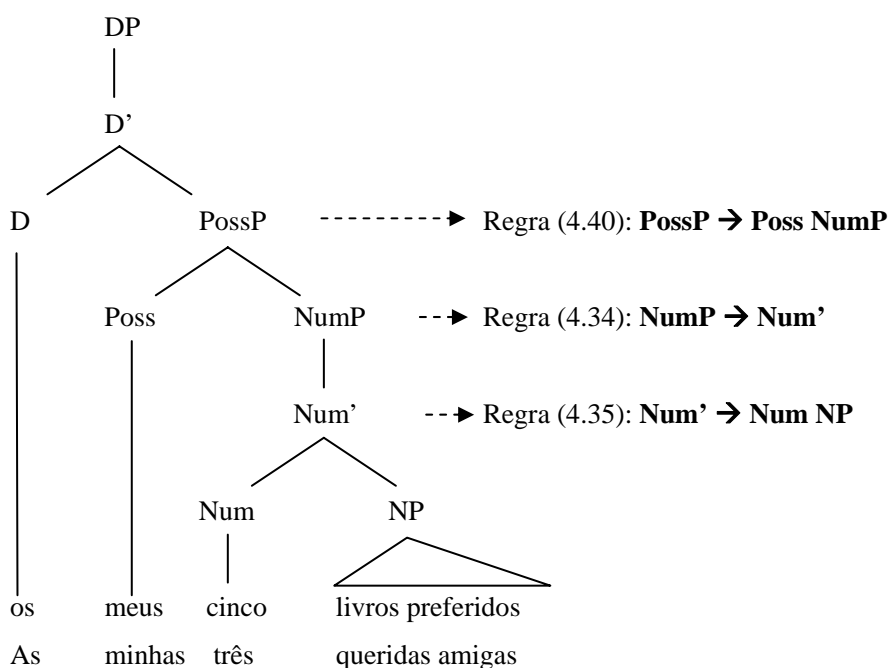


Como sugerimos acima, a restrição que irá permitir a boa formação de (4.90) e (4.93) e excluir (4.91) e (4.92) pode estar relacionada à implementação de um traço lexical [ $\pm$  **definido**]. Isso significa que as regras que estamos propondo parecem estar bem formuladas. Elas estão, na verdade, à espera da interação com o componente lexical, que irá trazer as informações necessárias sobre as categorias morfossintáticas e seu relacionamento dentro do DP, algo como propusemos na implementação da gramática do Grammar Play (cf. cap. 3), ou como fazem algumas abordagens gramaticais baseadas na unificação ou checagem de traços.

Passemos à última regra envolvendo os sintagmas PossP e NumP: a regra (4.40). Ela irá descrever estruturas de DPs como os seguintes:

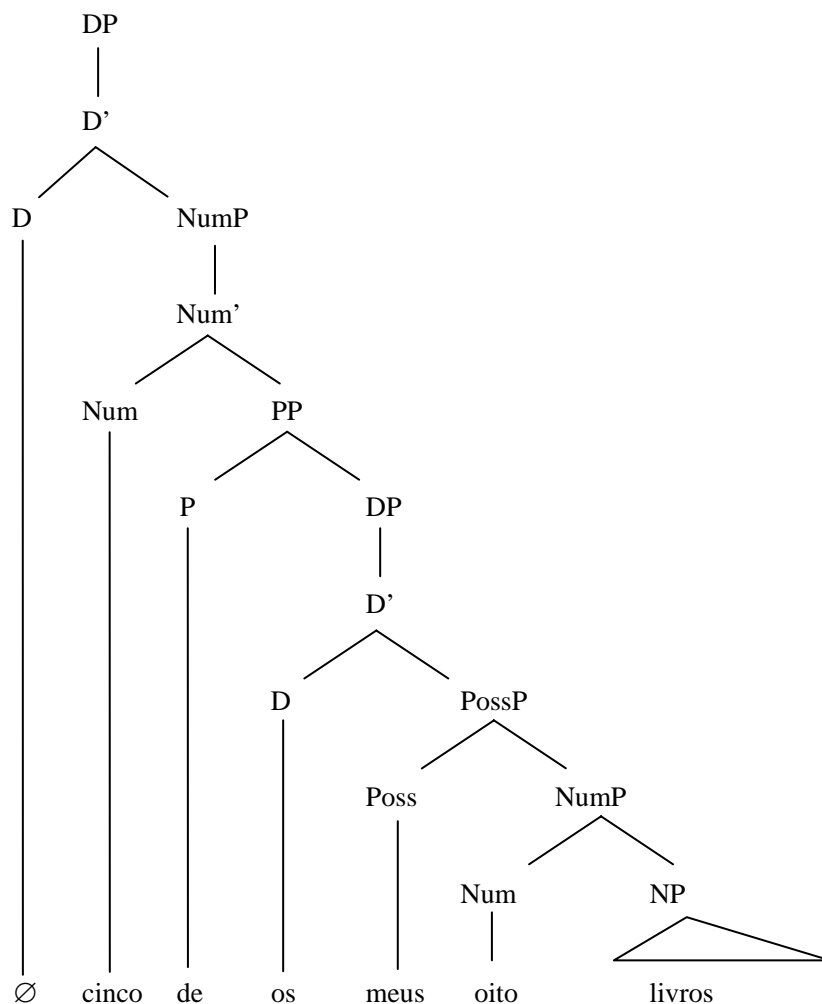
(4.96) A Maria leu [os meus cinco livros preferidos].

(4.97) [As minhas três queridas amigas] acabam de chegar.



A interação entre as regras dos elementos **Num** e **Poss** permitirá analisarmos corretamente sintagmas como (4.98), com um NumP dominando um PossP dominando um NumP, de maneira recursiva:

(4.98) [Cinco dos meus oito livros] desapareceram<sup>68</sup>.



Finalmente, passemos às análises envolvendo a projeção máxima QP. Adotamos o QP seguindo outras propostas e análises envolvendo sintagmas com quantificadores, como Shlonsky (1991), Raposo (1992), Radford (1997a,b), Vicente (2006), entre outros. Há na literatura diversos argumentos em favor da hipótese de se postular um núcleo funcional QP. Apresentaremos dois problemas de natureza sintática que parecem se beneficiar da postulação de um sintagma quantificador para análises em português.

O primeiro diz respeito a uma restrição em PB, conhecida como a **restrição do pronome pleno**. Esse fenômeno só acontece em línguas pro-drop (como o português)<sup>69</sup>.

<sup>68</sup> No entanto, sempre que lidamos com recursão, estamos lidando com um mecanismo extremamente poderoso das línguas e – até hoje – de difícil tratamento computacional. O problema aqui é que essas regras supergeram estruturas. Não há nada que impeça a formação do sintagma claramente inaceitável [**cinco de meus oito de meus dez de meus vinte livros**], por exemplo.

<sup>69</sup> Cf. Montalbetti (1984), Rothman & Iverson (2007), Molsing (2008).

Em línguas pro-drop, um pronome pleno não pode servir como elemento anafórico de um QP, como vemos no contraste das sentenças em português e inglês (língua não-pro-drop) abaixo:

(4.99) [Todos os alunos]<sub>i</sub> disseram que eles<sub>\*i/j</sub> são inocentes.

(4.100) [All the students]<sub>i</sub> said they<sub>i/j</sub> are innocent.

Em PB, é impossível ligar um pronome pleno a um QP. No entanto, essa restrição desaparece quando o referente é um simples DP:

(4.101) [Os alunos]<sub>i</sub> disseram que eles<sub>i/j</sub> são inocentes.

Postulando a distinção entre um QP e um DP, percebemos que a restrição do pronome pleno se dá, em PB, apenas com QPs. A restrição não atua sobre DPs, apontando para o fato de que eles são efetivamente de natureza distinta<sup>70</sup>.

Um outro argumento em favor da ideia do QP em português vem da análise de Costa (1997, 1998), quando ele mostra algumas diferenças no comportamento de sujeitos pré-verbais definidos e indefinidos. Para Costa (1998: 115), “diferentemente dos sujeitos pré-verbais definidos, os sujeitos pré-verbais indefinidos não podem co-ocorrer com extração de elementos-wh”, por motivos de restrições no movimento A-barrá<sup>71</sup>. Como exemplos, temos os seguintes pares:

(4.102)

a. Tu sabes quais livros a mulher leu \_\_\_?

b. \*Tu sabes quais livros uma mulher leu \_\_\_?

(4.103)

a. Que filme o menino alugou \_\_\_?

<sup>70</sup> Lembre-se de que o mesmo servirá para DPs contendo NumPs e PossPs, como (i) e (ii). Vimos que essas projeções estão contidas no DP; logo, são DPs. Repare como (i) e (ii) são gramaticais:

i. [Dois alunos]<sub>i</sub> disseram que eles<sub>i/j</sub> são inocentes.

ii. [Meus alunos]<sub>i</sub> disseram que eles<sub>i/j</sub> são inocentes.

Essa é mais uma evidência em favor de nossas regras gramaticais, que postulam que os NumPs e os PossPs são, de fato, DPs e que os agrupamentos com quantificadores são de outra natureza.

<sup>71</sup> Cf. argumentação em Costa (1997, 1998). Trecho original: “Differently from preverbal definite subjects, preverbal indefinite subjects may not cooccur with wh-extraction”. Sobre movimento A-barrá, cf. capítulo 6 de Culicover (1997), por exemplo.

b. \*Que filme um menino alugou \_\_?

O julgamento de gramaticalidade que distingue sentenças com movimento-wh com sujeitos pré-verbais definidos e indefinidos parece corroborar o argumento de Costa. No entanto, ele percebe que os julgamentos são afetados quando os sujeitos são DPs com elementos como *vários* (cf. Costa, 1998: 116):

(4.104) ??? Quais livros várias pessoas leram \_\_?

Ora, o fato que não foi percebido por Costa é que aqui não temos um DP, mas um QP<sup>72</sup>. E isso, por algum motivo, torna o julgamento de gramaticalidade mais confuso. A interpretação que sugerimos é que referencialmente um QP parece estar entre um DP definido e um DP indefinido, o que nos leva à escala de graus de gramaticalidade apontados por Costa:

(4.105) Quais livros [o João] leu \_\_?

(4.106) ? Quais livros [vários/todos os alunos] leram \_\_?

(4.107) \*Quais livros [um homem] leu \_\_?

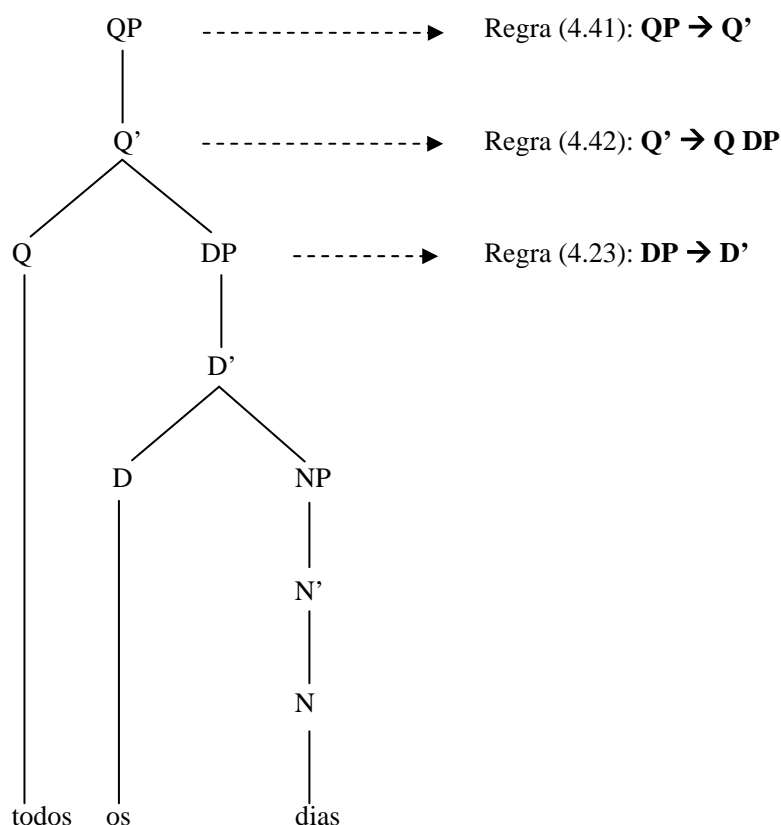
O fato que queremos apontar aqui é que uma análise levando em consideração o QP poderia esclarecer estes dois fenômenos em português: a restrição do pronome pleno e a extração de elementos-wh com sujeitos pré-verbais.

Agora, passemos à aplicação de nossas regras (4.41) a (4.44), relativas à estrutura interna do QP. Começemos com as seguintes análises:

(4.108) Ela sai [todos os dias].

---

<sup>72</sup> De fato, de acordo com Raposo (1992: 138), “em português, a classe dos Quantificadores inclui elementos como **muito(s)**, **pouco(s)**, **todo(s)**, **um**, **vários**, **bastantes**, entre outros” (grifos do autor).



Como as regras do **QP** não influenciam na estrutura interna do **DP**, a mesma regra permite tanto a análise de um sintagma como **[todos os dias]** quanto um sintagma como **[todos dias]**, este último com um determinante vazio. A análise da estrutura de um **DP** com o núcleo foneticamente vazio é a seguinte:  $QP[Q'[Q[todos]DP[D'D[\emptyset]NP[...]]]]$ . Um dos muitos problemas que iremos enfrentar com os quantificadores reside justamente aí, no fato de que alguns quantificadores exigem **DPs** com o **D vazio** (como *alguns*); outros com o **D preenchido** (como *ambos*); enquanto outros, como vimos com *todos*, são **indiferentes** em relação à presença explícita do determinante.

(4.109)

- a. [Alguns alunos] não fizeram o tema.
- b. \*[Alguns os alunos] não fizeram o tema.

(4.110)

- a. [Ambos os linguistas] chegaram à mesma conclusão.
- b. ? [Ambos linguistas] chegaram à mesma conclusão<sup>73</sup>.

<sup>73</sup> Essa sentença é perfeitamente gramatical para muitos falantes.

































































































































































