SINTAXE X-BARRA: UMA APLICAÇÃO COMPUTACIONAL

Sérgio de Moura MENUZZI (Universidade Federal do Rio Grande do Sul)
Gabriel de Ávila OTHERO (Pontifícia Universidade Católica do Rio Grande do Sul)

RESUMO: Neste trabalho, apresentaremos uma aplicação computacional da teoria X-barra (cf. Haegeman 1994, Mioto et al. 2004), através do programa Grammar Play, um *parser* sintático em Prolog. O Grammar Play analisa sentenças declarativas simples do português brasileiro, identificando sua estrutura de constituintes. Sua gramática é implementada em Prolog, com o recurso das DCGs, e é baseada nos moldes propostos pela teoria X-barra. O parser 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.

PALAVRAS-CHAVE: Teoria X-barra; Sintaxe Computacional; Processamento do Português.

***ABSTRACT**: In this article, we present an application of X-bar syntax in a computational enviroment. We present the parser Grammar Play, a syntactic parser in Prolog. The parser analyses simple declarative sentences of Brazilian Portuguese, identifying their constituent structure. The grammar is implemented in Prolog, making use of DCGs, and it is based on the X-bar theory (Haegeman 1994, Mioto et al. 2004). The parser is an attempt to broaden the coverage of similar syntactic analyzers, as the ones presented in Pagani (2004) and Othero (2006). The main goals of the present version of the Grammar Play are not related to broad coverage, but to the computational implemenation of coherent linguistic models applied to the description of*

Portuguese, and to the developement of a computational linguistics tool that can be used didactically in introductory classes of Syntax or Linguistics.

KEYWORDS: *X-bar Theory; Computational Syntax; Automatic Processing of Portuguese.*

0 Introdução

Neste artigo, estudaremos um pouco do processamento sintático computacional do português. Faremos isso de uma maneira prática, apresentando o programa Grammar Play, um *parser* sintático em Prolog para a língua portuguesa. Sua versão atual analisa sentenças declarativas simples do português brasileiro, identificando sua estrutura de constituintes, com base em uma gramática sintagmática implementada na linguagem Prolog. A gramática do *parser* foi implementada em Prolog com o recurso das DCGs, e é baseada no modelo de descrição proposto, em grande parte, pelo esquema da teoria X-barra *standard* (cf. Haegeman (1994) Mioto et al. (2004) para introdução e referências).

Em outras palavras, pretendemos apresentar um trabalho de descrição sintática das sentenças simples (aquelas que contêm apenas um verbo) do português, com o objetivo de implementação na linguagem computacional Prolog. Aqui, nos baseamos na análise apresentada em Othero (2006), para mostrar o desenvolvimento de uma gramática facilmente implementável em linguagem Prolog, com intuito de desenvolver o *parser* Grammar Play.

1 *Parsing* e Sintaxe Computacional

O estudo da estrutura sintática das línguas naturais tem origens muito antigas, que nos remetem a trabalhos de gramáticos gregos e latinos, como Apolônio Díscolo (gramático grego do século II d.C.), Donato e Prisciano (gramáticos romanos, dos séculos IV e VI d.C., respectivamente). O próprio termo *sintaxe* vem do grego *sýntaxis*, que é formado por *sýn* (junto) e *taxis* (ordenar, colocar). E a palavra *parsing* em si não remete ao processamento sintático mediado por computador (ou processamento

sintático computacional). O termo vem da expressão latina *pars orationes* (partes-do-discurso) e tem suas raízes na tradição clássica.

Em Lingüística Computacional, entretanto, *parsing* diz respeito à interpretação automática (ou semi-automática) de sentenças de linguagem natural por meio de programas de computador conhecidos como *parsers*. Esses programas são capazes de classificar morfossintaticamente as palavras e expressões de sentenças em uma dada língua e, principalmente, de atribuir às sentenças a sua estrutura de constituintes, baseando-se em um modelo formal de gramática.

De acordo com Covington (1994: 42), fazer o *parsing* de uma sentença é "determinar, por um processamento algorítmico, se a sentença é gerada por uma determinada gramática, e se ela for, qual estrutura que a gramática atribui a ela"[1]. Para Bateman, Forrest & Willis (1997: 166),

> um dos principais objetivos da área de PLN [Processamento de Linguagem Natural] 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[2]. (grifos dos autores).

Vários *parsers* já foram desenvolvidos nesses últimos nove dos anos, porém nenhum deles foi ainda capaz de alcançar o objetivo proposto por Bateman, Forrest & Willis. De acordo com Bick (2006), estes são os *parsers* e *taggers* (anotadores automáticos) atualmente disponíveis para a língua portuguesa (lista adaptada de Bick (2006)):

a) Curupira: *parser* baseado no formalismo da *phrase structure grammar*[3].

b) FreP: analisador fonológico no nível da palavra[4].

---

[1] Trecho original: "(...) to determine, by an algorithmic process, whether the sentence is generated by a particular grammar, and if so, what structure the grammar assigns it".

[2] 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.

[3] Cf. Martins, Hasegawa & Nunes (2002) e www.nilc.icmc.usp.br/nilc/tools/curupira.html.

[4] Cf. http://www.fl.ul.pt/LaboratorioFonetica/frep/.

c) GojolParser: baseado nos formalismos da *dependency grammar* e PSG[5].

d) Grammar Play: *parser* sintático, com gramática em Prolog. Utiliza o recurso das DCGs. Gramática baseada em PSG, seguindo proposta do esquema X-barra[6].

e) Hermes: *tokenizador* e *part-of-speech tagger*[7].

f) Jspell: analisador morfológico[8].

g) LX-Suite: lematizador, *PoS tagger* e *parser* sintático[9].

h) Palavras: um *parser* robusto baseado nos formalismos da gramática categorial, da DG e da PSG[10].

i) PoSiTagger: *PoS tagger* simbólico[11].

j) Q-tag: etiquetador morfológico automático para o português brasileiro[12].

k) TreeTagger: *PoS tagger*[13].

l) Xerox PoS tagger: *PoS tagger*[14].

Não é nosso intuito, no presente artigo, comparar estes vários aplicativos de processamento automático do português – especialmente aqueles dedicados ao *parsing* sintático. A tarefa, no entanto, é obviamente pertinente, e pretendemos enfrentá-la no futuro.

1.1 Grammar Play

O *parser* Grammar Play surgiu inicialmente como uma proposta de implementar em Prolog um analisador sintático que efetuasse o *parsing* de sentenças do português brasileiro com base em regras sintagmáticas que obedecessem o formato apresentado pelo esquema da teoria X-barra. Desde o princípio, o objetivo central com o desenvolvimento do Grammar Play foi o de implementar teorias lingüísticas consistentes para a descrição do PB em ambiente computacional que pudesse ser

---

[5] Cf. http://www.linguateca.pt/Repositorio/GojolParser.txt.
[6] Cf. Othero (2006) e http://www.geocities.com/gabriel_othero/public.html.
[7] Cf. http://hermes.sourceforge.net/hermesweb.html.
[8] Cf. http://natura.di.uminho.pt/natura/natura?&topic=jspell.
[9] Cf. http://lxsuite.di.fc.ul.pt/.
[10] Cf. Bick (2000) e http://visl.hum.sdu.dk/itwebsite/port/portgram.html.
[11] Cf. Aluísio & Aires (2000) e http://www.nilc.icmc.usp.br/nilc/projects/mestradorachel.html.
[12] Cf. http://lael.pucsp.br/corpora/etiquetagem/.
[13] Cf. http://gramatica.usc.es/~gamallo/tagger.htm.
[14] Cf. http://www.xrce.xerox.com/competencies/content-analysis/demos/portuguese.

utilizado didaticamente (em cursos introdutórios de teoria sintática ou de lingüística, por exemplo). O Grammar Play, tal como o apresentamos aqui, é uma tentativa de ampliação e revisão de propostas como as de Dougherty (1994), de Othero (2004) e de Pagani (2004).

Em sua versão atual, o Grammar Play analisa unicamente sentenças simples declarativas do PB, ou seja, sentenças que contenham apenas um verbo e não sejam interrogativas[15]. Ele é capaz de dizer quais dessas sentenças são gramaticais e quais não são, bem como é capaz de atribuir-lhes uma estrutura sintagmática baseando-se no modelo da teoria X-barra. 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).

Além de limitar-se à análise das sentenças simples do PB, o Grammar Play ainda não apresenta um componente morfológico e, por isso, pode analisar apenas sentenças que apresentam verbos conjugados na terceira pessoa do singular e do plural, no tempo presente do modo indicativo. Sabemos que isso limita consideravelmente o desempenho do *parser* em análises de textos naturais e espontâneos em PB. No entanto, enfatizamos que, até este momento, nosso objetivo maior com o desenvolvimento do Grammar Play não foi o de disponibilizar um programa robusto de análise morfossintática de textos naturais do PB; antes, queríamos iniciar a *implementação computacional de regras coerentes lingüisticamente,* baseadas em hipóteses teóricas relativamente consensuais em teoria lingüística e em resultados aceitos da descrição sintática do PB[16].

2 Regras Sintagmáticas, Gramática e Léxico

2.1 As regras sintagmáticas

[15] Na verdade, estamos deixando de lado apenas as interrogativas-QU, do tipo *Quem é amigo da Maria? O que a Maria quer de presente?* etc. Analisar frases interrogativas simples que não alteram sua estrutura com relação à sua correspondente declarativa não apresenta problemas. Compare os 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?*.

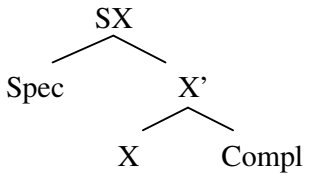
[16] Cf. Othero (2006), para mais detalhes.

As regras sintagmáticas de descrição da sentença simples em português brasileiro (PB) foram elaboradas dentro dos moldes propostos pela teoria X-barra. No quadro 1, apresentaremos algumas das regras que implementamos em Prolog, na gramática do Grammar Play:

Quadro 1: Regras sintagmáticas do Grammar Play

(1) SN → det N'
(2) SN → N'
(3) N' → N
(4) N' → SAdj N'
(5) N' → N' SP
(6) SAdj → Adj' SP
(7) SAdj → Adj'
(8) Adj' → Adj
(9) SP → P SN
(10) SP → P SAdv
(11) SV → V'
(12) SV → V' SP
(13) V' → V' SN
(14) V' → V
(15) V' → V SN
(16) V' → V SP
(17) SAdv → Adv'
(18) Adv' → Adv
(19) S → SN SV
(20) S → SV SN

Veja que o Grammar Play recorre a regras específicas a categorias (temos regras para o SN, para o SV, para o N', etc.), embora todas as versões da teoria X-barra proponham que as regras sintagmáticas são, na verdade, aplicações de esquemas gerais de organização sintagmática, esquemas estes que são "neutros" com relação à categoria do núcleo do sintagma (ver especialmente Stowell (1980) para discussão). O esquema geral de representação sintática proposto pelo modelo X-barra é o seguinte, adaptado de Mioto et al. (2004: 47):



No esquema acima, **X** pode ser um núcleo lexical de qualquer categoria (N, V, P, Adj ou Adv)[17]. Assim, a implementação que fazemos da teoria X-barra no *Grammar Play* segue seu espírito em alguns aspectos – como, por exemplo, procura oferecer uma mesma análise estrutural básica para os diferentes sintagmas, é binária, é endocêntrica[18] –, mas não é, ainda, uma "implementação de X-barra em Prolog". Muito provavelmente uma implementação "completamente de acordo" com o espírito da teoria X-barra – que se baseia na idéia de que "a sintaxe é resultado da projeção de propriedades lexicais" – exija um *parser bottom-up*, enquanto o Grammar Play segue técnicas mais tradicionais de *parsing top-down*.

2.2 O léxico

O léxico que utilizamos na gramática do Grammar Play é baseado em um corpus escrito de língua portuguesa disponível na Internet, no *site* do NILC (Núcleo Interinstitucional de Lingüística Computacional)[19], o corpus DELAS_PB. Esse corpus contém aproximadamente 67.500 palavras simples da língua portuguesa, listadas em sua forma canônica e separadas em diferentes categorias (substantivos próprios, substantivos comuns, advérbios, adjetivos e verbos).

Como nosso objetivo no desenvolvimento do Grammar Play não era produzir um analisador robusto, decidimos "enxugar" o corpus, mantendo apenas suas palavras "mais conhecidas". Para identificar as "palavras mais conhecidas", adotamos um critério bastante prático: pedimos a um falante nativo leigo de PB, de nível relativamente alto de escolaridade (estudante universitário), para que selecionasse apenas as palavras que ele conhecia daquele corpus. Isso acabou nos deixando, como resultado, com um léxico de cerca de 13.000 palavras (cerca de 4.000 verbos, 4.000 substantivos, 2.500 adjetivos, 2.500 advérbios e 18 preposições)[20].

Para trabalhar com o léxico em conjunto com as regras do Grammar Play, partimos da proposta original de implementação em Prolog apresentada por Pagani

---

[17] Não falaremos em núcleos funcionais aqui (como Infl, Comp, Agr...), já que o Grammar Play "reconhece" apenas os sintagmas de núcleo lexical SN, SV, SP, SAdj e SAdv).
[18] Com exceção, obviamente, da **S**, que não tem núcleo na leitura do Grammar Play.
[19] www.nilc.icmc.usp.br.
[20] Cf. Othero (2004 e 2006) para mais detalhes sobre o léxico do Grammar Play.

(2004), mais tarde revista por Othero (2004 e 2006). Nos quadros abaixo, apresentamos exemplos do tratamento que demos a diferentes itens lexicais no Grammar Play:

Quadro 2: Substantivos no léxico do Grammar Play

```
n([fem,sing], amiga).
n([fem,plur], amigas).
n([masc,sing], amigo).
n([masc,plur], amigos).
```

Quadro 3: Adjetivos no léxico do Grammar Play

```
% Adjetivos
adj([fem,sing], alta).
adj([fem,plur], altas).
adj([masc,sing], alto).
adj([masc,plur], altos).
```

Quadro 4: Determinantes no léxico do Grammar Play

```
% Pré-determinantes (pre_det)
pre_det([masc,plur], todos).
pre_det([fem,plur], todas).

% Determinantes-base (det)
det([masc,sing], o).
det([fem,sing], a).
det([masc,plur], os).
det([fem,plur], as).
```

Veja que, nos quadros 2, 3 e 4, a implementação dos itens lexicais nas categorias de substantivo, adjetivo e determinante envolve a codificação de certos aspectos gramaticais dessas classes por meio de termos do Prolog: cada entrada lexical contém uma lista com um termo codificando um valor de gênero (**masc** ou **fem**) e outro codificando um valor de número (**sing** ou **plur**). Esses termos serão responsáveis pela concordância nominal dentro do SN, pois fornecem o valor para as variáveis respectivas de número e gênero nas regras sintagmáticas. Da mesma maneira, os termos referentes a número são também responsáveis pela concordância verbal.

Veja alguns exemplos de implementação de preposições:

Quadro 5: Preposições no léxico do Grammar Play

```
% Preposições (p)
p(com, inf, com).
p(de, [_,_], de).
p(de, [masc, sing], do).
p(de, [fem, sing], da).
p(de, [masc, plur], dos).
p(de, [fem, plur], das).
```

A implementação apresentada no Quadro 5 foi proposta originalmente por Pagani (2004). Repare que essa interpretação analisa as preposições em dois tipos: há as preposições "inflexionáveis", marcadas pelo valor **inf**, e há as "flexionáveis", que são aquelas que se apresentam aglutinadas com o determinante. Nessa análise, as preposições flexionáveis são representadas no léxico como uma forma que já incorpora o determinante aglutinado e, a exemplo dos substantivos, adjetivos e determinantes, apresentam o valor correspondente do determinante para a flexão em gênero (**masc** e **fem**) e número (**sing** e **plur**). Pagani (2004: 20) sugere que essa estratégia de classificação das preposições é justificada:

> Esse tipo de solução para a distribuição das preposições sugere que existem dois grandes subconjuntos de preposições: as que são completamente inflexionáveis e as que aceitam flexão. Esse segundo subconjunto, por sua vez, se subdivide em outros dois subconjuntos: aquelas marcadas explicitamente com a flexão nominal e aquelas que não são marcadas, mas que, por isso mesmo, são compatíveis com qualquer concordância.

Entretanto, o tratamento acima indicado também implica que SNs regidos por "preposições flexionáveis" não terão a mesma "análise sintagmática" que outros SNs – o determinante aglutinado à preposição *não* é analisado como parte do SN. Evidentemente, isso também não se conforma ao espírito da teoria X-barra, nem às análises sintáticas correntes do PB e, por isso, este aspecto da implementação deve ser foco de avaliação nas versões futuras do Grammar Play. No presente, a grande vantagem da proposta de Pagani é prática: permite-nos implementar a análise de SPs como *dos meninos* sem que precisemos de uma análise morfossintática do item *dos*.

No quadro 6, apresentamos a implementação dos verbos na gramática do Grammar Play:

Quadro 6: Verbos no léxico do Grammar Play

```
% Verbos Intransitivos
v(sing, i, morre).
v(plur, i, morrem).

% Verbos Transitivos Diretos (td)
v(sing, td, come).
v(plur, td, comem).

% Verbos Transitivos Indiretos (ti)
v(sing, ti(de), precisa).
v(plur, ti(de), precisam).

% Verbos Bitransitivos (tdi)
v(sing, tdi(em), coloca).
v(plur, tdi(em), colocam).

% Verbos de Ligação (cop)
v(sing, vl, continua).
v(plur, vl, continuam).
```

Todos os verbos apresentam, em sua entrada lexical, um valor correspondente à informação de número (**sing** ou **plur**). Já vimos que os substantivos também apresentavam esses mesmos valores. Esses valores, como já havíamos mencionado, serão responsáveis pela concordância verbal por meio da unificação com as variáveis correspondentes nas regras sintagmáticas do sintagma verbal e da sentença. Talvez, mereça comentário aqui o fato de que, embora estejamos assumindo como base a teoria X-barra, não estamos assumindo outros aspectos técnicos dos modelos gerativistas mais difundidos no Brasil. Por exemplo, nossa implementação do processo de concordância faz uso de mecanismos de unificação de traços, encontrados em teorias pouco difundidas no Brasil, como a *Head Driven Phrase Structure Grammar* (ver, por exemplo, Bender, Sag & Wasow (2003)), e não por via transformacional e checagem de traços, como normalmente assumido em modelos chomskyanos (como pressuposto em Mioto et al. (2004) e Haegeman (1994)).

Repare, ainda com relação às entradas lexicais dos verbos, que há nelas um outro termo, correspondente a um valor que chamamos de *valência*. O valor de valência é correspondente à variável **Val** nas regras sintagmáticas do SV, variável que permite unificar a valência de um verbo com a sua regência, ou esquema de subcategorização, concretamente expressa na sentença. Essa idéia de implementação foi originalmente proposta por Pereira & Shieber (1987).

O Grammar Play identifica cinco diferentes tipos de verbos, de acordo com sua valência: verbos intransitivos (**i**), verbos transitivos diretos (**td**), verbos transitivos indiretos (**ti**), verbos bitransitivos (**tdi**) e verbos de ligação, ou cópula, (**cop**). Esses valores funcionam como índices que apontam para regras sintagmáticas específicas, cuja estrutura corresponde à regência exigida pelo verbo, ou seja: há uma regra para SVs intransitivos, outra para SVs transitivos diretos, etc. Essa implementação, obviamente, não segue o espírito "categorialmente neutro" proposto pela teoria X-barra. Antes, está mais próxima do modelo padrão da gramática gerativa (Chomsky (1965)), que foi eliminado gradualmente em função da massiva redundância entre a informação lexical e a informação constante nas regras sintagmáticas. É claro que uma implementação computacional mais fiel de modelos gramaticais correntes deve fazer jus a esta idéia fundamental.

Por fim, consideremos a implementação dos advérbios. Como são palavras invariáveis, suas entradas lexicais, diferentemente das dos substantivos, verbos, etc., não apresentam atributos que atuem em processos de concordância:

Quadro 7: Advérbios no léxico do Grammar Play

```
adv(abertamente).
adv(agora).
adv(nunca).
```

3 Implementação das regras em Prolog

Para começarmos o processo de implementação das regras em Prolog na gramática do Grammar Play, partimos da descrição sintagmática da sentença simples em PB descritas por Othero (2004 e 2006). Para implementação, fizemos uso do recurso das DCGs, um formalismo de representação de gramáticas livres de contexto. Esse recurso torna fácil a implementação de uma gramática sintagmática em Prolog, uma vez que "uma gramática descrita em uma DCG é diretamente executada pelo Prolog como um analisador sintático" (Bratko, 1997: 431).

Vejamos como implementamos algumas das regras sintagmáticas do Grammar Play. No quadro 8, tratamos das regras (1) a (5), que dizem respeito à boa formação da sentença, do sintagma verbal e do sintagma nominal (as regras aqui implementadas também aparecem no quadro 1).

Quadro 8: Implementação das regras sintagmáticas no Grammar Play

```
. Regras sintagmáticas:
(1)  SN → det  N'
(2)  N' → N
(3)  SV → V'
(4)  V'→ V  SN
(5)  S → SN  SV

.Implementação das regras em Prolog:
(1')  sn(Conc, [det, [Det], n_bar, N_Barra]) --> det(Conc, Det), n_barra(Conc, N_Barra).
(2')  n_barra(Conc, [n, [N]]) --> n(Conc, N).
(3')  sv(Num, [v_bar, V_Barra]) --> v_barra(Num, _, V_Barra).
(4')  v_barra(Num, td, [v, [V], sn, SN]) --> v(Num, td, V), sn(_, SN).
(5')  s([sn, SN, sv, SV]) --> sn([_,Num], SN), sv(Num, SV).
```

Vejamos que estratégias foram adotadas na implementação das regras sintagmáticas; em particular, que informações as regras de nosso *parser* codificam além daquelas que constam em (1)-(5).

A regra (1') diz que um **SN** apresenta uma variável **Conc**, que deverá ser unificada com as variáveis **Conc** do determinante e do N-barra ao lado direito da regra: essas variáveis eventualmente serão unificadas com os valores de gênero e número fornecidos pelas entradas lexicais do determinante e do substantivo, assegurando a concordância entre os constituintes do SN, e entre o SN sujeito e o verbo (concordância de número). Além disso, a regra (1') verifica, evidentemente, a constituição sintagmática do SN: deve ser formado por um determinante, expresso pela variável **Det**, e por um N-barra, isto é, um constituinte nominal intermediário, expresso pela variável **N'** (em Prolog, **N_Barra**).

Já a regra (2') diz que o **N'** é formado, no mínimo, por um **N** (um substantivo); outras regras de expansão do N' permitem a introdução de modificadores do N (como sintagmas adjetivais, sintagmas preposicionais, etc.). Além disso, novamente aparece a variável **Conc**, que, como já mencionamos, deve unificar-se com os valores de gênero e número especificados na entrada lexical do **N**.[21]

A regra (3') diz que um **SV** pode ser formado apenas por um constituinte verbal intermediário, **V'**. (Outras regras permitem que o SV contenha modificadores, como sintagmas adverbiais, sintagmas preposicionais, etc.) Além disso, a regra (3') diz que o SV apresenta uma variável **Num**, que deve ser unificada com a variável **Num** do V', que também apresenta um valor de número, expresso pela mesma variável, **Num**.

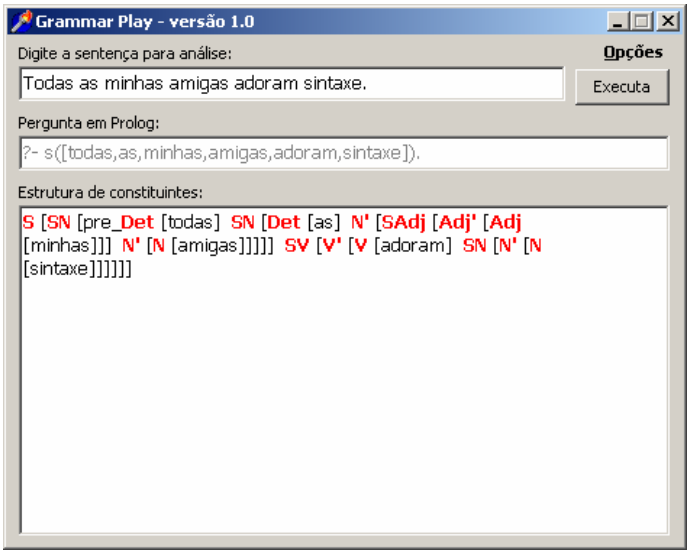
[21] Aqui estamos omitindo as regras de inserção lexical presentes na gramática do *parser*. A regra para inserção lexical do N, por exemplo, é **n(Conc, N) --> [N], {n(Conc, N)}.**

Ambas as variáveis serão, eventualmente, unificadas com o valor correspondente encontrado na entrada lexical do verbo núcleo do SV. Além disso, a variável **Num** do SV deverá ser unificada com a correspondente encontrada no SN sujeito, como se vê na regra (5'), de formação da sentença. Esse último processo de unificação é responsável pela **concordância verbal** no Grammar Play.

Note que, além de um valor associado à variável **Num** encontrada no SV, mencionamos antes que as entradas lexicais dos verbos possuem ainda um valor de **valência**, responsável pela subcategorização verbal. Na verdade, pela regra (3'), não podemos perceber os efeitos do valor de valência verbal — de fato, o lugar da variável correspondente é ocupado por um traço de *underline* (isto é, pela chamada "variável anônima" do Prolog). Por que isso? Simplesmente porque não nos interessa saber que tipo de verbo (intransitivo, transitivo direto, bitransitivo...) pode formar um V' que formará um SV, já que todos eles podem.

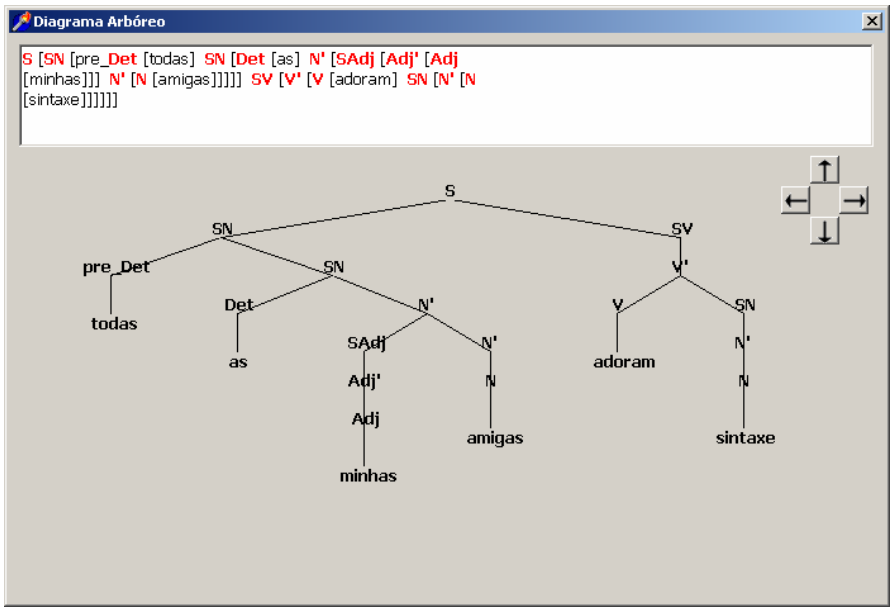
A regra (4') traz o valor **td** para a valência. Isso quer dizer que essa regra se aplica somente ao V' cujo núcleo é um verbo transitivo direto. Ou, em outros termos, o Grammar Play autoriza que um verbo forme um V' com um SN complemento somente se o verbo possui, também, em sua entrada lexical o valor **td** para valência. Com base nessas cinco regras, pode-se analisar sentenças como *Todas as minhas amigas adoram sintaxe*. De acordo com nossas regras, análise dessa sentença é a seguinte (as figuras são da interface gráfica do Grammar Play):

Figura 1: Todas as minhas amigas adoram sintaxe – colchetes rotulados



E a árvore sintática da sentença pode ser vista na figura 2:

Figura 2: Todas as minhas amigas adoram sintaxe – estrutura arbórea



5 Considerações finais

Vimos que o Grammar Play não pretende ser um *parser* robusto para a língua portuguesa. Ele pretende ser uma aplicação computacional de teorias sintáticas correntemente utilizadas para a descrição da língua portuguesa. Ainda assim, como tentamos mostrar ao longo deste texto, a gramática do *parser* nem sempre é adequada para o modelo que gramática que adotamos: uma gramática sintagmática baseada nos moldes propostos pela teoria X-barra. Contudo, acreditamos que o Grammar Play possa servir como uma ferramenta útil e interessante em cursos de Lingüística Formal, Lingüística Computacional, Sintaxe e Processamento do Português: suas próprias deficiências práticas e conceituais servem para indicar os rumos em que a pesquisa em implementação computacional de modelos teóricos deve perseguir: sempre em busca de maior eficiência, mas também maior compatibilidade com o conhecimento corrente dos modelos gramaticais.

Referências Bibliográficas

ALUÍSIO, S. M.; AIRES, R. V. *Etiquetação de um corpus e construção de um etiquetador de português*. Trabalho técnico, 2000.

BATEMAN, J.; FORREST, J.; WILLIS, T. The use of syntactic annotation tools: partial and full parsing. In: GARSIDE, R.; LEECH, G.; McENERY, A. *Corpus annotation: linguistic information from computer text corpora*. London / New York: Longman, 1997.

BENDER, E. M.; SAG, I. A.; WASOW, T. *Syntactic theory: a formal introduction.* Stanford: CSLI Publications, 2003.

BICK, E. *The parsing system Palavras - automatic grammatical analysis of Portuguese in a constraint grammar framework*. Aarhus: Aarhus University Press, 2000. PhD Thesis.

BICK, E. Automatic syntactic annotation. Trabalho apresentado durante a *Primeira Escola de Verão da Linguateca*. Porto, 2006.

BRATKO, Ivan. *Prolog programming for artificial intelligence*. Harlow: Addison-Wesley, 1997.

CHOMSKY, N. *Aspects of the theory of syntax*. Cambridge: MIT Press, 1965.

COVINGTON Michael. A. *Natural language processing for Prolog programmers*. New Jersey: Prentice Hall, 1994.

DOUGHERTY, R. C. *Natural language computing: an English generative grammar in Prolog*. Hillsdale: Lawrence Erlbaum, 1994.

HAEGEMAN, L. *Introduction to government and binding theory*. Oxford: Blackwell, 1995.

MARTINS, R. T.; HASEGAWA, R.; NUNES, M. G. V. *Curupira: um parser funcional para o português*. NILC-TR-02-26, Dezembro de 2002.

MIOTO, C. et al. *Novo manual de sintaxe*. Florianópolis: Insular, 2004.

OTHERO, G. A. *Grammar Play: um* parser *sintático em Prolog para a língua portuguesa*. Porto Alegre: PUCRS, 2004. Dissertação de Mestrado.

OTHERO, G. A. *Teoria X-barra: descrição do português e aplicação computacional*. São Paulo: Contexto, 2006.

PAGANI, L. A. Analisador gramatical em Prolog para gramáticas de estrutura sintagmática. *Revista Virtual de Estudos da Linguagem – ReVEL*, ano 2, n. 3, agosto de 2004. [http://paginas.terra.com.br/educacao/revel/index.htm]

PEREIRA, F.; SHIEBER, S. M. *Prolog and natural-language analysis*. Stanford: CSLI, 1987.

STOWEL, T. A. *Origins of phrase structure*. Massachusetts: MIT, 1981. PhD Thesis.

Anexo - Trabalhando com o Grammar Play

Nesta seção, mostraremos rapidamente como funciona o Grammar Play (ver nota 6 para o sítio de *download* do programa). Para fazer o programa operar, basta escrever um frase no campo indicado e clicar no botão "executa". O programa irá convertê-la para notação em Prolog, no campo "pergunta", e dará sua análise da estrutura de constituintes da sentença através da representação de colchetes rotulados, como na figura 3. Para obter a representação em árvore, basta clicar em "Opções" e selecionar "Representação arbórea", e o resultado será como na figura 4.

Figura 3: Análise de "O João gosta da Maria" – colchetes rotulados

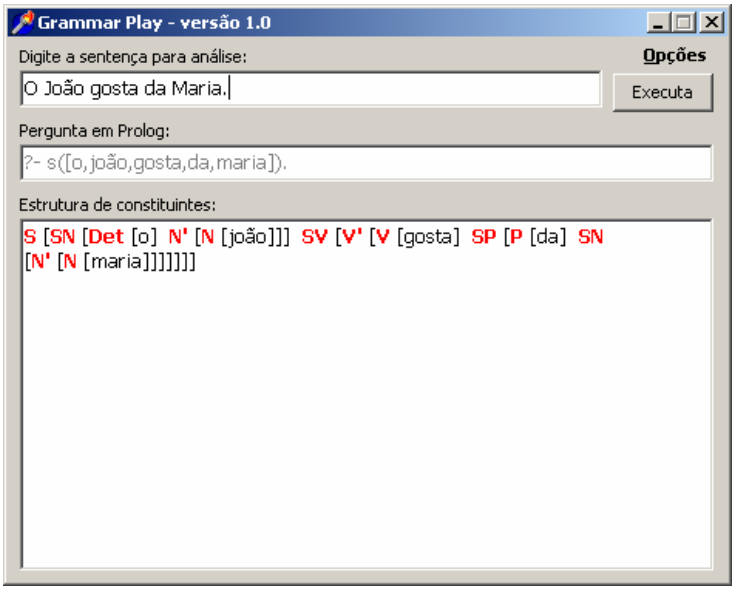
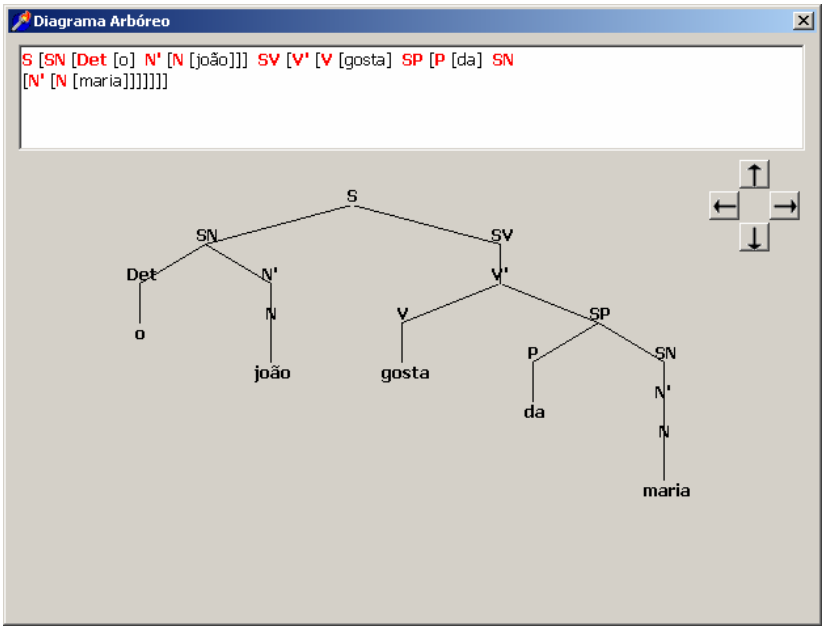


Figura 4: Todos os meus amigos adoram sintaxe – estrutura arbórea



Se o Grammar Play não reconhecer a frase como válida na língua, quer por ela ser agramatical (fig. 5), quer por limitações que o programa ainda apresente, ele retornará a seguinte mensagem: “frase agramatical ou léxico desconhecido”.

Figura 5: *As todas adoram amigas minhas sintaxe