ALENCAR, Leonel Figueiredo. Resenha de "Teoria X-barra: descrição do português e aplicação computacional", de Gabriel de Ávila Othero. *Revista Virtual de Estudos da Linguagem – ReVEL*. Vol. 6, n. 10, março de 2008. ISSN 1678-8931 [www.revel.inf.br].

RESENHA DE "TEORIA X-BARRA: DESCRIÇÃO DO PORTUGUÊS E APLICAÇÃO COMPUTACIONAL", DE GABRIEL DE ÁVILA OTHERO

Leonel Figueiredo de Alencar[1]

prof_leonel-sintaxe@yahoo.com.br

À Editora Contexto se deve esse que é um dos lançamentos mais importantes no mercado editorial brasileiro do ano passado, no setor de manuais direcionados ao público universitário das áreas de lingüística e informática. Trata-se, pelo que sabemos, do primeiro livro a focar, no Brasil, uma dimensão da gramática gerativa pouco explorada entre nós, mas bastante consolidada na Europa e nos EUA, que é a sua aplicabilidade na informática, mais especificamente na construção de analisadores sintáticos. Esses programas, tecnicamente chamados de *parsers*, partem de uma dada gramática, enquanto definição de uma língua formal, para determinar se uma dada expressão pertence a essa língua e atribuir a cada construção gramatical uma ou mais representações estruturais. Para que isso funcione, contudo, a descrição gramatical fornecida ao *parser* deve ser especificada num rigoroso formalismo matemático, de forma totalmente explícita. Uma descrição, nesses moldes, de um expressivo fragmento do português do Brasil é o que nos oferece Gabriel de Ávila Othero, com base num dos pilares do modelo Princípios e Parâmetros da gramática gerativa: a teoria X-barra.

Embora se situe na interseção entre lingüística e informática, só pela contribuição à descrição do português esse trabalho já merece destaque, uma vez que,

[1] Professor do Programa de Pós-Graduação em Lingüística e do Departamento de Letras Estrangeiras da Universidade Federal do Ceará.

nessa área, predominam estudos não formais ou semi-formais.[2] A relação entre gramática gerativa e ciência da computação, no entanto, não se limita à mera aplicação da primeira no âmbito dessa última. Constitui, na verdade, uma via de mão dupla. Uma compreensão mais aprofundada da relevância desse livro só é possível, ao nosso ver, a partir de uma visão do quadro interdisciplinar amplo em que se insere.

Qual a utilidade das pesquisas em gramática gerativa? Por que utilizar a matemática, algo tão identificado com as ciências naturais, para descrever um objeto que, aparentemente, pertence à diametralmente oposta esfera da cultura? São questionamentos freqüentes entre estudantes de Letras, às voltas com as famosas regras de reescrita, fórmulas e árvores que, num público (infelizmente) pouco afeito à matemática, levam a uma rejeição imediata do gerativismo. A conclusão apressada é a de que, nesse paradigma, o método de representação empregado simplesmente não se harmoniza com o objeto de estudo.[3]

Seguindo um raciocínio completamente diferente, lingüistas que trabalham sob a perspectiva gerativa sustentam que apenas a formalização da análise de uma língua em particular, e das propriedades da Linguagem em geral, permite expressar com precisão hipóteses sobre esses objetos de estudo e testá-las rigorosamente, uma vez que isso pode ser feito de forma automática, sem recurso à intuição, sujeita a falhas (Sag, Wasow & Bender 2003:8). Tudo isso com o objetivo maior não tanto de fundamentar alguma metodologia de ensino de línguas ou a escolha de conteúdos nesse domínio, mas, sobretudo, de observar um dos aspectos mais fascinantes do funcionamento da mente humana (Sag, Wasow & Bender 2003:9).

Disso decorre outra diferença de pensamento fundamental entre os indevidamente chamados "formalistas" e (se nos permitem o neologismo) os "informalistas", refratários à modelação matemática de teorias lingüísticas: a Linguagem integra o mundo natural, como parte que é do cérebro, cujas estruturas são geneticamente determinadas (Raposo 1992:26). Logo, se a física, química e biologia

---

[2] Para Falk (2001:28-29), tanto na TRL quanto no Programa Minimalista (os dois principais modelos gerativos utilizados na descrição do português) predomina o uso de "prosa informal" para descrever regras e princípios, em detrimento de uma formalização "perfeitamente explícita" que, nas palavras do próprio Chomsky (1965:4), constitui a essência de uma gramática gerativa.

[3] Adotamos, aqui, a distinção de Schwarze (2001:5) entre *métodos de descoberta* e *métodos de representação*. Na lingüística brasileira, métodos de descoberta baseados na matemática têm sido bastante utilizados, há muitos anos, em pesquisas de variação lingüística e, mais recentemente, na lingüística de corpus. A matemática não goza, porém, da mesma popularidade no caso dos métodos de representação.

contemporâneas são impensáveis sem o recurso a modelos formais, o mesmo se aplica à lingüística.

Um número menor, mas expressivo, de lingüistas responderia, à questão sobre a utilidade da gramática gerativa, do seguinte modo: descrições informais, em si mesmas, são inúteis para o desenvolvimento de programas computacionais capazes de processar automaticamente algum aspecto da linguagem natural. Por outro lado, após meio século de pesquisas em gramática gerativa, podemos contabilizar vários exemplos de programas que, baseados nessa teoria lingüística, conseguem apontar erros ortográficos e gramaticais, traduzir com certa acuidade determinados tipos de textos ou interagir com seres humanos em linguagem natural.[4] É claro que essa tecnologia precisa ser aperfeiçoada, especialmente no que tange à língua portuguesa. Quem teve oportunidade de utilizar programas desse tipo para línguas como inglês ou alemão com certeza deve ter notado a defasagem que similares para o português apresentam. Isso não surpreende quando se considera que a lingüística computacional está presente, há várias décadas, em dezenas de departamentos de lingüística e computação na Europa e nos EUA, ao passo que é tão fracamente representada no Brasil, mesmo nos centros de ciências e tecnologia de nossas universidades. A discrepância de qualidade entre os aplicativos voltados para o português, de um lado, e para o inglês e alemão, de outro, mostra que o progresso na área é uma questão de tempo e de investimento em pesquisas envolvendo o tratamento computacional da linguagem natural.

Como ressaltamos no início, a lingüística computacional não constitui somente uma área de aplicação das pesquisas em gramática gerativa. O inverso também se verifica, com a primeira área constituindo ferramenta importantíssima na construção de modelos dessa última.[5] De fato, a tarefa do lingüista gerativo é muitas vezes descrita como a de descoberta das representações mentais e dos algoritmos executados pelos falantes sobre essas representações na produção ou recepção de frases.[6] É evidente que esse empreendimento será tão mais rigoroso, quanto seja possível testar empiricamente em computadores modelos computacionais da Linguagem construídos pelo lingüista.

---

[4] Para uma visão geral da área da tecnologia da linguagem, com avaliações sobre os progressos já alcançados nas diferentes áreas do processamento computacional da linguagem natural, ver Mitkov (2005).

[5] Sobre a contribuição da lingüística computacional para a lingüística teórica, ver, especialmente, Gómez (2000).

[6] A esse respeito, Carnie (2002:5), por exemplo, afirma: "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."

Talvez não seja exagero dizer que o estágio atual de sofisticação das teorias gerativas decorre, em grande parte, dos progressos na ciência da computação (e, evidentemente, nas disciplinas que informam essa ciência, como é o caso da matemática). Cabe ressaltar que a lingüística gerativa e a ciência da computação nasceram praticamente na mesma época (i.e., meados do século XX) e que Noam Chomsky não é só o fundador dessa teoria lingüística, mas está entre os cientistas que mais contribuíram para a teoria dos compiladores, por meio de estudos na área das línguas formais.

É diante desse amplo cenário que a publicação de Othero se revela extremamente oportuna. De fato, estabelece um paradigma a ser seguido, especialmente por estudantes de pós-graduação, que têm na interseção entre lingüística computacional e gramática gerativa, no que concerne à língua portuguesa, um vasto e promissor território de investigação científica ainda pouquíssimo explorado.

O livro resulta de sua dissertação de Mestrado em Lingüística Aplicada na PUCRS, sob orientação de Sérgio Menuzzi, que, no prefácio, com muita propriedade destaca se tratar "de uma obra importante no contexto da pesquisa brasileira, pois procura promover a união de tradições de pesquisa que, em nosso ambiente acadêmico, ainda cooperam pouco entre si – a Lingüística e as Ciências da Computação" (p. 14).

A partir de exemplos extraídos, sobretudo, de gramáticas e manuais introdutórios de sintaxe, Othero desenvolve um fragmento de gramática computacional do português capaz de reconhecer como gramaticais essas frases (ou construções estruturalmente equivalentes) e construir as respectivas representações arbóreas em termos de uma das primeiras versões da teoria X-barra. Esse fragmento foi elaborado no formalismo conhecido como Gramática de Cláusulas Definidas (doravante DCG, abreviatura de *Definite Clause Grammar*), que permite implementar na linguagem de programação Prolog, de modo relativamente fácil, gramáticas livres de contexto aumentadas com estruturas de traços. Essas gramáticas são processadas em Prolog por um algoritmo de parsing que opera sobre o input da esquerda para a direita e de cima para baixo. Pela operação matemática de unificação, os traços permitem modelar fenômenos como concordância e regência de forma direta, sem recorrer a transformações como movimento e apagamento de traços nem a princípios como, por exemplo, o da regência na TRL. Atualmente, a DCG, criada no início dos anos 80, está superada por formalismos de unificação não só mais expressivos como também mais robustos e eficientes tais como a LFG e HPSG. A mais conhecida limitação do algoritmo utilizado por Prolog para processar DCGs é que não admite regras com

recursividade à esquerda, do tipo que encontramos, por exemplo, em configurações de adjunção, como no exemplo (1).

(1) V' → V' PP

No entanto, em que pese essa limitação, inexistente em sistemas que a sucederam (por exemplo, PCPATR, ver Black 1997), a DCG, devido exatamente a sua simplicidade formal, ainda é ensinada em muitas universidades de ponta nos EUA e na Europa em cursos introdutórios de lingüística computacional.[7]

Além do prefácio, ao qual já nos referimos, e de sucintas, mas muito bem elaboradas introdução e conclusão, o livro compõe-se de três capítulos principais e um apêndice. O primeiro capítulo, de maior interesse para informatas, expõe a metodologia utilizada na construção do fragmento de gramática computacional. O segundo capítulo, de relevância imediata para todo estudioso da sintaxe, discute algumas das principais análises gerativas tradicionais dos sintagmas nominal, adjetival, adverbial, preposicional e verbal do português do Brasil e propõe análises alternativas no formato X-barra. Finalmente, no terceiro capítulo, é descrita passo a passo a implementação, no formalismo DCG, das regras formuladas no capítulo anterior.

No seu sítio na Internet, o autor disponibiliza gratuitamente, além de exercícios sobre cada capítulo do livro, um programa que criou, no âmbito do Mestrado, chamado GrammarPlay, cujo uso é explicado no apêndice. Trata-se de uma interface gráfica que simplifica bastante a análise sintática automática de frases conforme a gramática desenvolvida ao longo do livro, na medida em que dispensa a interação por linha de comando com um compilador (ou interpretador) da linguagem de programação Prolog, não imediatamente acessível a não programadores.

Desse modo, de um ponto de vista estritamente lingüístico, o livro é útil para o estudante de sintaxe sem conhecimentos de programação que quer testar seus conhecimentos da versão da teoria X-barra enfocada. Como o autor contemplou no analisador sintático exemplos de abordagens tradicionais e de versões mais antigas da gramática gerativa, esse programa pode ser uma ferramenta bastante útil no estudo da sintaxe gerativa, pois o usuário pode comparar diferentes análises de uma mesma construção com a análise baseada nessa versão da teoria X-barra.

[7] Um exemplo é a disciplina *Introduction to Computational Linguistics* da Havard University, ministrada por Suart M. Shieber no semestre de outono 2005-2006.

Quem trabalha na área de sintaxe computacional familiariza-se, logo de início, com a noção de fragmento. Como uma descrição sintática computacional deve ser sempre totalmente formalizada, um analisador automático é necessariamente limitado de algum modo, em termos de cobertura. Essas limitações, para as quais o autor chama atenção, provêm tanto do tratamento do léxico quanto da sintaxe propriamente dita.

De fato, para que a análise automática de uma frase gramatical seja feita com sucesso, é preciso que todos os itens lexicais estejam descritos detalhadamente no léxico. Essa descrição não pode se limitar a uma mera listagem dos lexemas, mas deve incluir os paradigmas flexionais de cada um deles e suas respectivas propriedades morfossintáticas. Isso é indispensável para a modelação computacional de fenômenos como concordância e regência. Dadas as limitações de tempo próprias do projeto que originou o livro, um componente morfológico que gerasse os paradigmas flexionais não pôde ser implementado.

A construção de um analisador sintático sem um componente morfológico que fizesse jus à riqueza da morfologia flexional do português implicaria a listagem, como ressalta o autor (p. 35), de algo em torno de 59 formas para cada verbo. Desse modo, para simplificar, ele se limitou, na construção do léxico, às formas da 3ª pessoa do presente do indicativo. Compensando essa limitação, o léxico é relativamente vasto quanto ao número de lexemas contemplados.

No âmbito da sintaxe, o autor se limitou à descrição da frase simples. Mesmo nesse domínio, porém, a descrição se limita às estruturas mais básicas. Por exemplo, a gramática elaborada por Othero leva em conta estruturas como (2) e (3), com o quantificador na posição canônica dentro do sintagma nominal. No primeiro caso, o quantificador é analisado como "pré-determinante". No segundo, é analisado como adjetivo.

(2) Todas as meninas dormem tranqüilamente.

(3) As meninas todas dormem tranqüilamente.

A dupla categorização de *todas* nesses exemplos é indesejável em gramática gerativa, mas tolerável num programa de análise sintática automática. O problema maior é que estruturas como (4), porém, com um "stranded quantifier", não são contempladas.

(4) As meninas dormem todas tranqüilamente.

Essas limitações, porém, não invalidam o grande esforço que o autor empreendeu, esforço esse que resultou no analisador sintático do português do Brasil baseado na gramática gerativa mais abrangente, pelo que sabemos, atualmente disponível de forma gratuita.[8] Uma ampla cobertura das variações da estruturação sintática de uma língua como o português só poderia ter sido alcançada no âmbito de um projeto de equipe, desenvolvido durante muitos anos e utilizando modelos lingüísticos computacionais mais sofisticados e técnicas de indução automática do léxico e da sintaxe.

O que Othero oferece, portanto, é um fragmento de gramática do português do Brasil que contempla um léxico relativamente extenso e as estruturas mais básicas de uma sintaxe X-barra tradicional. Apesar de se tratar de um fragmento, contudo, essa gramática constitui um excelente ponto de partida, por um lado, para projetos de estudantes na área de sintaxe computacional, que podem, de diferentes maneiras, expandir o fragmento já constituído, de modo a dar conta, por exemplo, de fenômenos como a flutuação de quantificadores. Por outro lado, lingüistas computacionais podem, com proveito, tomá-la como esqueleto inicial na construção de ferramentas mais sofisticadas de análise sintática computacional do português.

Othero denomina a versão da teoria X-barra utilizada de *padrão*. De fato, há hipóteses nessa teoria que se tornaram padrão na sintaxe gerativa, no sentido de mais amplamente difundidas, como a ramificação binária, o número máximo de duas barras e a distinção entre especificador, complemento e adjunto conforme, respectivamente, as regras (5), (6) e (7), na formulação de Radford (1988:277).[9]

(5) X'' → X', (YP)
(6) X' → X, YP*
(7) X' → X', YP

[8] Um analisador, disponível livremente, mais robusto que o GrammarPlay é o VISL Portuguese, elaborado por Eckhard Bick (URL <http://visl.sdu.dk/visl/pt/>). Esse sistema, porém, não se baseia na gramática gerativa. Embora contemple um leque muito mais amplo de estruturas sintáticas e lexicais, esse analisador paga um preço excessivo pela robustez, hipergerando de forma inaceitável. Por exemplo, frases como **Todos as meninas todas dorme todas tranqüilamente* e **Dormem meninos os* são analisadas como gramaticais.

[9] Observe que a regra (6) viola o binarismo.

Por outros aspectos, porém, o termo padrão é inadequado como caracterização do modelo sintático que subjaz ao fragmento gramatical de Othero, até porque está calcado em Chomsky (1970), a primeira formulação da teoria X-barra. Expressões nominais como *as meninas*, por exemplo, não são tratadas como DPs (o que se tornou padrão em sintaxe gerativa em meados da década de 80 com os trabalhos de Fukui e Abney), mas, na esteira de Chomsky (1970), como sintagmas nominais em que o determinante funciona como especificador. A denominação do modelo adotado como *teoria X-barra tradicional*, portanto, teria sido mais feliz.[10]

Em vários outros aspectos Othero desvia-se do que de fato é comum a várias versões da teoria X-barra. Limitamo-nos a quatro exemplos. Em primeiro lugar, ele propõe, como uma das regras da frase simples, a tradicional fórmula S → SN SV. Ora, essa regra foge ao esquema da teoria X-barra, violando o princípio da endocentricidade, esse sim uma hipótese que se tornou padrão em sintaxe gerativa e resume como nenhuma outra o espírito da teoria X-barra. Aparentemente, não há justificativa para adotar essa análise da frase simples, quando já Chomsky (1986:3) propôs que S, na verdade, é projeção máxima da categoria flexão, i.e. I" ou IP.

Em segundo lugar, o tratamento dos adjuntos (tanto adverbiais quanto adnominais) em Othero é bastante idiossincrático. Como vimos em (7), na teoria X-barra que podemos classificar como padrão, adjuntos são categorias não nucleares cujo irmão e pai são idênticos, i.e. categorias que "expandem um constituinte de um tipo noutro constituinte do mesmo tipo" (Radford 1988:255). O autor, contudo, propõe as regras tanto de (8) quanto de (9) para dar conta de constituintes como o destacado em (10).

(8) SV -> Sadv V' (p. 70)

(9) V' -> Sadv V' (p. 70)

(10) [Atualmente lá já existem] alguns tratores. (p. 80)

A regra (9), porém, já dá conta desse constituinte. A regra (8), "teoricamente incorreta" como reconhece o próprio Othero (porque trata o Sadv como especificador), justificar-se-ia, segundo ele, por permitir construir "uma árvore menos complicada de

[10] Consulte-se, a esse respeito, Grewendorf (2002, 33-35), que denomina de *estrutura X-barra tradicional* a abordagem da teoria X-barra imediatamente anterior a hipóteses como a de Split-INFL e do sujeito interno ao VP.

ver" (p. 73). A regra (9), contudo, em conjunção com a regra que expande SV em V' (p. 70), incorre no mesmo tipo de problema.

Em terceiro lugar, o que poderíamos chamar de teoria padrão distingue claramente, como vimos, complemento e especificador, por um lado, de adjunto, por outro. Apenas adjuntos são filhos e irmãos de uma categoria de mesmo tipo que funciona como núcleo (Grewendorf, Hamm & Sternefeld 1989:207). Othero, porém, considera o segundo complemento de verbos bitransitivos como filho e irmão de V', numa configuração, portanto, de adjunção. É verdade que essa "heresia" à teoria X-barra tradicional tem sido praticada por vários autores, para salvar a ramificação binária e/ou o número máximo de duas barras.[11] No entanto, como ressalta Grewendorf (2002:61), é "conceptualmente problemático" postular que um argumento de um verbo seja gerado numa posição de adjunção. De fato, essa é a posição de argumentos movidos (Grewendorf 2002:34). Cremos, porém, que, para ser coerente com a abordagem tradicional de Chomsky (1970), que fundamenta, em linhas gerais, o modelo sintático de Othero, teria sido melhor sacrificar a ramificação binária, como fazem Radford (1988) (ver (6)) e o próprio Chomsky (1986).

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

(11) *Todas ambas ambas todas ambas as meninas dormem tranqüilamente.

Em conclusão, pode-se afirmar que, não obstante essas deficiências, se trata de trabalho muito bem estruturado, redigido e editado (com mínimos erros de revisão), didático, bem fundamentado e, antes de tudo, de uma honestidade exemplar, na medida em que dificuldades e problemas não resolvidos não são "varridos para debaixo do tapete", mas claramente expostos pelo autor, estimulando pesquisas futuras.

[11] Black (1997:8) e Berman (2003:37), por exemplo, adotam análises, em linhas gerais, nesses moldes.

Consideramos a obra, portanto, de aquisição obrigatória não só para informatas e lingüistas computacionais interessados no processamento computacional do português do Brasil, mas também para todo interessado em sintaxe, especialmente estudiosos da gramática gerativa. Apesar de pressupor, no terceiro capítulo, algum conhecimento de Prolog, o livro é amplamente acessível a não-programadores, uma vez que o programa GrammarPlay, explicado no final da obra, permite analisar automaticamente os exemplos do livro sem precisar construir comandos de Prolog.

REFERÊNCIAS

1. BERMAN, Judith. 2003. *Clausal Syntax of German*. Stanford: CSLI.
2. BLACK, Cheryl A. 1997. A PC-PATR implementation of GB syntax. *SIL Electronic Working Papers*. **6**. Disponível em:<http://www.sil.org/silewp/1997/006/> Acesso em: 29.11.2006.
3. CARNIE, Andrew. 2002. *Syntax: A Generative Introduction*. Oxford: Blackwell.
4. CHOMSKY, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge: MIT.
5. _____. 1970. Remarks on Nominalization. In: R. A. Jacobs & P. Rosembaum. Eds. *Readings in English Transformational Grammar*. Waltham: Ginn.
6. _____. 1986. *Barriers*. Cambridge: MIT.
7. FALK, Yehuda N. 2001. *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*. Stanford: CSLI Publications.
8. GÓMEZ, Xavier Guinovart. 2000. Lingüística computacional. In: Fernando Ramallo, Gabriel Rei-Doval & Xoán Paulo Rodríguez. Eds.. *Manual de Ciencias da Linguaxe*. Vigo: Xerais. p. 221-268.
9. GREWENDORF, Günther, Fritz Hamm & Wolfgang Sternefeld. 1989. *Sprachliches Wissen: Eine Einführung in moderne Theorien der grammatischen Beschreibung*. Frankfurt am Main: Suhrkamp.
10. _____. 2002. *Minimalistische Syntax*. Tübingen & Basel: A. Francke.
11. MITKOV, Ruslan. Ed. 2005. *The Oxford Handbook of Computational Linguistics*. Oxford: Oxford University Press.
12. RADFORD, Andrew. 1988. *Transformational Grammar: A First Course*. Cambridge: Cambridge University Press.

13. RAPOSO, Eduardo Paiva. 1992. *Teoria da Gramática: A Faculdade da Linguagem*. Lisboa: Caminho.
14. SAG, Ivan A., WASOW, T.; BENDER, E. *Syntactic Theory: A Formal Introduction*. 2. ed. Stanford: CSLI Publications, 2003.
15. SCHWARZE, Christoph. 2001. *Introduction à la Sémantique Lexicale*. Tübingen: Narr.

OTHERO, Gabriel de Ávila. *Teoria X-barra: descrição do português e aplicação computacional*. São Paulo: Contexto, 2006.