

GRAMÁTICAS CATEGORIAIS, CONCORDÂNCIA E COMPUTADORES

José Borges Neto*

Márcio Renato Guimarães**

O “problema”

Existe hoje disponível no mercado uma série de *softwares* que são capazes de fazer, com alto grau de eficiência, a correção ortográfica de textos escritos. Mais difíceis de achar, no entanto, são programas capazes de fazer a correção gramatical dos textos,¹ mesmo se pensamos em coisas bastante simples, como concordância e regência, por exemplo.

Pensemos particularmente na concordância. Sabemos que há uma concordância dita *nominal* e outra dita *verbal*. A concordância nominal consiste numa relação entre um substantivo e as palavras que o cercam, como adjetivos, artigos, numerais, etc. Essa relação consiste numa “equivalência de traços”, i.e.,

* Universidade Federal do Paraná.

** Unicentro - Guarapuava, PR.

1 A equipe do Núcleo Interdepartamental de Linguística Computacional (NILC) da USP de São Carlos, coordenada pelos professores Osvaldo Oliveira Júnior e Maria das Graças V. Nunes, está desenvolvendo, em parceria com a Itaotec-Philco, um revisor gramatical bastante eficiente, capaz de corrigir, entre outras coisas, a concordância. Trata-se, no entanto, de programa ainda em desenvolvimento e, até onde sabemos, ainda não disponível comercialmente.

se o substantivo apresenta o “traço” masculino, esse “traço” também tem que estar presente no artigo, no adjetivo, etc. Os “traços” pertinentes à concordância nominal seriam referentes a *gênero* (masculino/feminino) e *número* (singular/plural). A concordância verbal consiste na relação entre um sintagma nominal (o “sujeito”) e um sintagma verbal (o “predicado”). Como na concordância nominal, trata-se também aqui de “equivalência de traços”: o “sujeito” e o “predicado” precisam apresentar os mesmos valores para os traços de *número* (singular/plural) e *pessoa* (primeira, segunda ou terceira).

O “problema”, então, seria o de desenvolver um mecanismo que pudesse ser implementado de forma a fazer o computador (o editor de textos, para ser mais claro) reconhecer que em certos lugares do texto há um “erro de concordância”.

Alguém poderia perguntar: e esse seria uma problema *lingüístico*? A resposta, cremos, seria: sim e não. Se houvesse, disponíveis, descrições lingüísticas razoáveis do processo de concordância nominal,² por exemplo, o programador não teria problemas maiores no desenvolvimento do *software* (os problemas que encontraria seriam problemas de programação). Como não existem descrições razoáveis da concordância nominal, esse problema passa a ser também um problema do lingüista.

Para quem está pensando “Bolas! O que há de complicado na concordância nominal? As gramáticas escolares já listam todos os casos particulares, além da regra geral. A descrição do fenômeno já está feita!”, só poderíamos responder: essa descrição não serve. Infelizmente, dizer para um computador que “o adjetivo concorda em gênero e número com o substantivo” é absolutamente inútil.³

É tarefa do lingüista, então, prover o profissional de computação de descrições adequadas dos fenômenos lingüísticos, para que este possa realizar seu trabalho. Há, no entanto, dois modos distintos de fazer isso: um modo dirigido pelas necessidades do profissional da computação e um modo “lingüisticamente orientado”.

No primeiro modo, levantam-se os “erros” de concordância nominal que são normalmente cometidos⁴ e procuram-se meios de corrigi-los. O programador

2 E quando dizemos *razoáveis* queremos dizer, entre outras coisas, *suficientemente formalizadas*.

3 O interessante é que provavelmente dizer isso para os alunos de português de primeiro ou segundo grau é inútil também.

4 Obviamente não vamos entrar em nenhuma discussão sociolingüística a respeito da noção de erro. Trata-se aqui de correções de desvios de uma língua padrão adequada ao texto escrito, e cremos que ninguém discordaria da maioria dos “erros” apontados nesse

está interessado em tornar o seu programa o mais simples possível e, portanto, vai querer evitar todo tipo de mecanismo que se destine a resolver casos que jamais aparecerão. Por exemplo, suponhamos que o que se “erra” na concordância nominal seja normalmente o número – erros de concordância de gênero seriam estatisticamente mais raros. O programador vai dar atenção maior, então, para os processos de concordância de número e, eventualmente, pode até ignorar determinados casos de concordância de gênero. As descrições obtidas desse modo serão sempre parciais e “especializadas” – elas são construídas especificamente para a realização de uma tarefa. Obviamente, se conduzida de forma adequada, a descrição dos fenômenos lingüísticos feita desse modo permite obter “ferramentas” computacionais altamente eficientes, embora de baixo interesse lingüístico.

No segundo modo, o lingüista descreve o processo lingüístico sem pensar em utilizações específicas. Mas a descrição é feita de forma rigorosamente formal, de modo a poder ser computada. Nada fica subentendido. Todos os passos são explicitados. Tudo – o que é problema e o que não é problema; o que as pessoas “erram” e o que as pessoas não “erram” – tem que aparecer na descrição. Em outras palavras, a descrição feita pelo lingüista deve ser absolutamente *explícita*.⁵ Certamente, enquanto lingüistas, é deste modo que pretendemos tratar os fenômenos lingüísticos.

Vejamos como isso pode se dar, apresentando uma pequena descrição – tentativa – do processo de concordância nominal que (i) se pretende uma descrição *lingüística* e (ii) se pretende uma descrição lingüística *computável*. Antes de apresentar a descrição é preciso, no entanto, fazer uma breve introdução à teoria que a suporta.

A gramática categorial (GC)

O melhor modo de apresentar uma GC “clássica” é definir a noção de *categoria* e enunciar a *regra de aplicação funcional*, que é a sua única regra. Geralmente inicia-se com duas categorias básicas (ou “argumentais”), *N* (nome) e *S* (sentença), e definem-se as outras categorias (derivadas ou “funtoras”) por

quadro.

5 Era disso que Chomsky falava, já nos anos 50, quando dizia que a tarefa do

um mecanismo recursivo.⁶ O conjunto de categorias fica definido então pelas regras de (1):

- (1) a. N e S são categorias.
- b. Se α e β são categorias, então α/β é uma categoria (para quaisquer categorias, básicas ou derivadas, α e β).

A regra de aplicação funcional recebe a seguinte formulação:

- (2) Se A é uma expressão da categoria α/β e B é uma expressão da categoria β , então, a expressão AB (= concatenação das expressões A e B) é uma expressão da categoria α .

A especificação de uma GC para uma língua qualquer consistiria em apresentar uma lista das expressões básicas dessa língua associadas a símbolos de categorias. O conjunto de expressões “geradas” pela gramática (i.e., o conjunto de expressões “bem-formadas” da língua) é o conjunto de expressões básicas mais o conjunto de expressões obtidas pela regra de aplicação funcional.

As necessidades decorrentes das tentativas de aplicação às várias línguas naturais (e a algumas linguagens formais), levaram à proposta de várias extensões da GC “clássica”.⁷ Vamos tentar, no entanto, construir a descrição do processo de concordância nominal, o mais possível, no quadro da GC clássica.

Uma gramática categorial para o sintagma nominal (SN) do português

Como a concordância nominal se dá no interior do SN, não vamos nos preocupar com outros aspectos da construção da sentença em português, restringindo-nos aos limites do SN.

Suponhamos que as categorias sintáticas que se manifestam no interior

lingüista devia ser a construção de gramáticas *gerativas*, i.e., explícitas.

6 A idéia básica é de que as categorias são conjuntos de expressões (semelhantes às “classes de palavras” das gramáticas escolares) que possuem comportamento sintático e semântico uniforme. As categorias básicas são aquelas que funcionam na estrutura das sentenças sempre como *argumentos de funções*; as categorias derivadas, por sua vez, seriam aquelas que funcionariam como *funções*. Há várias fontes onde se pode ter acesso a exposições gerais, destinadas a leigos, sobre essas gramáticas, como, por exemplo, Borges Neto (1979) e Lyons (1979, cap. 6).

7 Por exemplo, novas categorias básicas, categorias direcionais, operadores combinatórios, novas regras além da regra de aplicação funcional, etc. Ver Wood (1993) para uma apresentação geral dessas extensões.

do SN sejam a categoria dos *nomes comuns* (*nc*), a categoria dos *adjetivos* (*adj*) e a categoria dos *quantificadores* (*qu*). É preciso ainda prever a categoria dos *sintagmas nominais* (*sn*), que seria a categoria geral das expressões sob análise.⁸ Dessas quatro categorias, suponhamos que a categoria dos *nc* e a categoria dos *sn* seriam categorias básicas.⁹ As outras categorias seriam derivadas. Os adjetivos seriam *nc/nc* (i.e., operadores que junto com um *nc* resultam em outro *nc*, conforme a regra de aplicação funcional), e os quantificadores seriam *sn/nc* (operadores que junto com um *nc* resultam em um *sn*).

Em todas as categorias, básicas ou derivadas, vamos encontrar *expressões básicas* (ou *atômicas*) e *expressões complexas*. Façamos uma pequena lista (parcial) de expressões básicas de nossas quatro categorias:

- (3) *adj* = {alta, alto, amarela, amarelo, bonita, bonito, generosa, generoso, verde, vermelha, vermelho}
nc = {bandeira, cobra, homem, inseto, menina, menino, muro, papel, parede, rã, sapo, velhote}
qu = {a, o, um, uma, toda, todo}
sn = {Ana, Beatriz, Carlos, Davi, Eliana... eu, você, ela, ele...}

Toda expressão básica da categoria *sn* é um *sintagma nominal*. Toda expressão básica da categoria *adj* concatenada com uma expressão, básica ou não, da categoria *nc* é um *nome comum*. Toda expressão básica da categoria *qu* concatenada com uma expressão, básica ou não, da categoria *nc* é um *sintagma nominal*. Vamos tentar formular isso numa linguagem mais rigorosa. Suponhamos que a operação que um adjetivo realiza ao se concatenar com um nome comum seja chamada de f_1 e que a operação que um quantificador realiza ao se concatenar com um nome comum seja chamada de f_2 . Assim, por f_1 , a expressão que consista na concatenação de um adjetivo e um nome comum pertencerá à categoria *nc*. Por f_2 , a expressão resultante da concatenação de um quantificador e um nome comum pertencerá à categoria dos *sn*. Dito formalmente, numa notação que ordena <operação, *input*, *output*>:

- (4) Regra 1 (a) $\langle f_{1a}, \langle adj, nc \rangle, nc \rangle^{10}$

8 Talvez seja necessário ampliar o número das categorias. Não queremos, no momento, tratar desse assunto.

9 Na realidade, estamos simulando aqui uma GC "clássica" que, em lugar de N e S, apresenta *nc* e *sn* como categorias básicas. Se estivéssemos tratando da sentença, a categoria S (sentença) seria uma categoria básica e a categoria *sn* seria derivada. Ao tratarmos apenas do sintagma nominal, a categoria *sn* passa a ser básica, já que ela não é parte de nenhuma outra categoria no fragmento.

$$\text{Regra 2} \quad (b) \quad \begin{array}{l} \langle f_{1b}, \langle nc, adj \rangle, nc \rangle \\ \langle f_2, \langle qu, nc \rangle, sn \rangle \end{array}$$

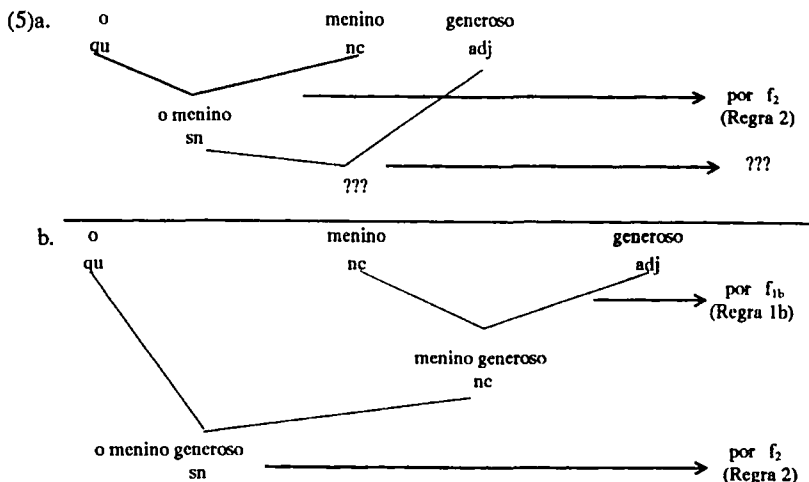
Trocando em miúdos, numa fórmula como $\langle f_2, \langle qu, nc \rangle, sn \rangle$ está-se dizendo que uma operação f_2 se dá sobre o par ordenado $\langle qu, nc \rangle$ e que, como resultado dessa operação, o par de expressões se torna uma “unidade” que é da categoria sn .

Vamos tentar imaginar como procederia um computador diante de alguns casos. Imaginemos que alguém escrevesse (digitasse!) *Carlos*. O computador localizaria a palavra *Carlos* na lista dos sn e registraria esse fato. Nada mais haveria para ser feito (uma expressão básica da categoria sn é um sn [$sn = sn$]). Imaginemos agora que alguém escrevesse *o menino*. O computador localizaria a palavra *o* na lista dos qu e a palavra *menino* na lista dos nc . Ora, a seqüência $qu\ nc$ é *input* possível para a operação f_2 , que ao se aplicar vai resultar numa expressão da categoria sn – *o menino* (uma expressão qualquer da categoria qu concatenada a uma expressão qualquer da categoria nc formam um sn [$qu + nc = sn$]). Por último, imaginemos que alguém escrevesse *o menino generoso*. Inicialmente, o computador localizaria as palavras nas listas e ficaria com uma seqüência de três categorias: $qu + nc + adj$. Essa seqüência permite uma dupla estruturação: $(qu + nc) + adj$ ou $qu + (nc + adj)$. A rigor, o computador deve considerar ambas as estruturações:¹¹ se ele seguir o primeiro caminho, concatenando inicialmente $qu + nc$, vai ficar com uma seqüência $sn + adj$ que não pode ser concatenada; se ele seguir o segundo caminho, concatenando inicialmente $nc + adj$, vai obter um nc que poderá ser concatenado em seguida com o qu para resultar num sn . Para ficar mais claro, podemos representar esse processo todo numa estrutura arborescente como a seguinte:

Esse tipo de mecanismo, extremamente simples,¹² permitiria gerar todos os sintagmas nominais “corretos” do fragmento do português. Permitiria tam-

10 Note-se que para fins de concordância o fato de o adjetivo anteceder ou seguir o nome comum é irrelevante. Como o adjetivo em português pode, em princípio, aparecer em ambos os lugares e não é clara a diferença entre um adjetivo anteposto e um posposto, estamos abrindo a possibilidade de tratar ambos de um mesmo e único jeito, considerando que a operação é a mesma nos dois casos.

11 O problema aqui é semelhante ao problema da ordem das operações numa expressão aritmética. Poderíamos, então, pensar em alguma “convenção” que ordenasse as operações (algo como f_1 se aplica antes de f_2) ou algum outro mecanismo que evitasse as ambigüidades “espúrias”. Temos a impressão, no entanto, de que a manutenção irrestrita das ambigüidades sintáticas pode ser linguisticamente interessante.



bém, no entanto, gerar vários sintagmas nominais “agramaticais”. Por exemplo, geraríamos “uma menino”, “o parede vermelho”, “a sapo verde”, etc. Justamente aí é que entram as regras de concordância que nos interessam.

A concordância de gênero no SN do português

Para dar conta da concordância de gênero nos sintagmas nominais do português precisamos ampliar nossa gramática em duas direções: precisamos sofisticar nosso léxico (as listas de palavras categorizadas) e precisamos introduzir um mecanismo de “checagem” de traços na regra de aplicação funcional. Começemos com o léxico.

Além do símbolo de categoria, as expressões básicas precisam ser associadas a “traços”: traços de gênero e número. Como fazer isso? Parece que a forma mais simples de fazer isso é entender a expressão no léxico como um “feixe de traços”. Em outras palavras, ao invés de termos o léxico organizado em listas, como fizemos acima, passamos a ter o léxico organizado como uma coleção de entradas lexicais complexas. Por exemplo, a lista dos nomes comuns, que em (3) era:

12 A rigor, próximo de um mecanismo de “ensaio-e-erro”.

- (6) $nc = \{ \text{bandeira, cobra, homem, inseto, menina, menino, muro, papel, parede, rã, sapo, velhote} \}$

passaria a ser enriquecida com subscritos, como em:

- (7) $nc = \{ \text{bandeira}_{F,S}, \text{cobra}_{F,S}, \text{homem}_{M,S}, \text{inseto}_{M,S}, \text{menina}_{F,S}, \text{menino}_{M,S}, \text{muro}_{M,S}, \text{papel}_{M,S}, \text{parede}_{F,S}, \text{rã}_{F,S}, \text{sapo}_{M,S}, \text{velhote}_{M,S} \}$
(onde M indica *masculino*, F indica *feminino* e S indica *singular*)

Alternativamente, podemos entender o léxico como uma seqüência de palavras em, digamos, ordem alfabética, cada uma associada a um feixe de traços (pensemos provisoriamente numa entrada em que apareçam, na ordem, a palavra, a categoria sintática da palavra, o gênero e o número). Algo como o seguinte:¹³

(8)

<i>a</i>	<i>alta</i>	<i>alto</i>	<i>amarela</i>	<i>Ana</i>
qu	adj	adj	adj	sn
F	F	M	F	F
S	S	S	S	S
<i>Beatriz</i>	<i>bonita</i>	<i>bonito</i>	<i>estudante</i>	<i>inteligente</i>
sn	adj	adj	nc	adj
F	F	M	α	α
S	S	S	S	S
<i>o</i>	<i>papel</i>	<i>verde</i>	<i>vermelhas</i>	<i>vermelhos</i>
qu	nc	adj	adj	adj
M	M	α	F	M
S	S	S	PL	PL

Obviamente, poderíamos também pensar em mecanismos morfológicos que tornassem mais enxuto esse léxico, evitando entradas distintas para masculino e feminino ou singular e plural (sempre que as formas fossem previsíveis por regras). Não vamos entrar em detalhes aqui sobre a arquitetura desse léxico, porque essa não é a questão que nos preocupa no momento.

¹³ Onde M é *masculino*, F é *feminino*, S é *singular*, PL é *plural* e α é uma *variável* que indica que o traço em questão não é especificado (algo como α = M ou F)

Existe uma versão específica da gramática categorial que se baseia inteiramente nessa definição de itens lexicais como feixes, ou matrizes, de traços: as gramáticas de unificação categoriais¹³ (doravante GUC). As GUC surgiram especificamente voltadas para a lingüística computacional. Elas diferem substancialmente de outras modalidades das gramáticas categoriais justamente no recurso aos traços, que substituem recursos utilizados por modelos alternativos – conjuntos de regras, por exemplo.

Uma GUC compõe-se unicamente de uma definição de categoria e de uma regra de aplicação. Outros modelos de gramáticas categoriais possuem um conjunto de regras mais sofisticado. As GUC, porém, se caracterizam por esse sistema de regras bastante sucinto. Tudo o que é resolvido em outros modelos de gramática categorial com regras, numa GUC é resolvido através de traços e unificação de traços.

As categorias, numa GUC, são definidas como uma matriz portando três traços fundamentais: valor, direção e argumento. Para o traço de direção, utilizam-se barras (/ ou \), ou alguma abreviação do tipo DIR ou ESQ, conforme a categoria peça um argumento à esquerda ou à direita, respectivamente. A definição de quantificadores, por exemplo, numa GUC, ficaria como em (9):

(9) Quantificadores

VALOR: SN					
DIREÇÃO: /					
ARGUMENTO: NC					
TRAÇOS:	<table style="border-collapse: collapse; border: none;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">GÊNERO</td> <td style="padding: 5px;"> </td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">NÚMERO</td> <td style="padding: 5px;"> </td> </tr> </table>	GÊNERO		NÚMERO	
GÊNERO					
NÚMERO					

Essa notação pode ser um tanto simplificada definida como uma maneira diferente de representar aquilo que acima representamos como frações. Lembre-se que os quantificadores foram definidos como operadores “fracionários” do tipo sn/nc ($\frac{sn}{nc}$), i.e., expressões que junto a um nc , resultam

14 O termo *Categorial unification grammars* foi inadequadamente traduzido para o português como *gramática categorial de unificação* (em Guimarães, 1997). Não é apenas uma questão de “ordem de tratores”. Wood (1993, p. 70) observa as marcantes diferenças que existem entre as *unification categorial grammars* (gramáticas categoriais de unificação), como a de Zeevat, Klein e Calder (1987) e as *categorial unification grammars* (gramáticas de unificação categoriais), como a de Bouma (1988), que serve de modelo para a análise apresentada aqui. Segundo Wood, as gramáticas de unificação categorial são apresentadas como um modelo geral que embasa teorias diferentes, que nem sempre concordam umas com as outras em seus princípios, como um formalismo que “incorpora as propriedades essenciais tanto da unificação quanto do formalismo das gramáticas categoriais” (Uzkoreit, 1986, p.187). Já a gramática categorial de unificação é uma teoria específica dentro desse modelo.

num *sn*. Podemos defini-los como funções, que tomam um *nc*, à direita, como argumento e dão como valor um *sn*. Valor e argumento correspondem, assim, ao numerador e ao denominador da fração, respectivamente.

Essa notação das categorias dispensa as regras expressas em (4). Uma vez que, dentro da matriz, já vem especificado que tipo de argumento o operador toma, e qual a direção em que esse argumento é localizado, não existe por que manter um sistema de regras de ordem de palavras dentro do SN.

Categorias simples (como os nomes e as sentenças) possuem apenas traços de valor. Traços de direção e de argumento aparecem apenas em categorias complexas. Isso é fácil de entender: como as categorias simples funcionam apenas como argumento de outras categorias, e não tomam argumento (verbos, p. ex., pedem sujeito e/ou objetos, mas substantivos não pedem necessariamente verbos), não há por que expressar esses traços.

Além desses três traços fundamentais, que vamos chamar aqui de *traços categoriais*, todas as matrizes comportam outros traços. Com relação a o que são e quais são esses “outros traços” não existe um consenso entre os autores que trabalham com gramáticas categoriais. Cada autor lista os traços que acha mais relevantes, ou que são mais relevantes para o problema específico de que está tratando. Como o nosso interesse é dar conta da concordância dentro dos *sn* do português, chamaremos esses traços de traços morfológicos.

Com relação à concordância nominal em português, a matriz deve especificar traços como os de número e gênero. Se estivéssemos analisando, ao invés do português, a concordância nominal em latim ou grego, a matriz deveria especificar também o caso:

(10) a. Português: *meninas*

VALOR: NC					
TRAÇOS:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">GEN: fem</td> <td style="padding: 5px;"> </td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">NUM: plural</td> <td style="padding: 5px;"> </td> </tr> </table>	GEN: fem		NUM: plural	
GEN: fem					
NUM: plural					

b. Latim: *puellas*

VALOR: NC							
TRAÇOS:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">GEN: fem</td> <td style="padding: 5px;"> </td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">NUM: plural</td> <td style="padding: 5px;"> </td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">CASO: acusativo</td> <td style="padding: 5px;"> </td> </tr> </table>	GEN: fem		NUM: plural		CASO: acusativo	
GEN: fem							
NUM: plural							
CASO: acusativo							

Só os traços que uma categoria pode receber, ou que são relevantes, são expressos. Assim, por exemplo, em português, os verbos têm flexões de número, pessoa, tempo/aspecto e modo, mas não têm flexão de gênero. Portanto, é relevante expressar, na matriz, apenas traços correspondentes a número/pessoa, tempo/aspecto e modo. No caso de línguas como o hebraico ou as línguas

eslavas, teríamos que expressar traços de gênero, pois nessas línguas o verbo apresenta também flexões de gênero.

Os modelos básicos de quantificadores, adjetivos e nomes comuns, em português, são expressos a seguir:

(11) a. Quantificadores

VALOR: SN	
DIREÇÃO: /	
ARGUMENTO: NC	
TRAÇOS:	GÊNERO
	NÚMERO

b. Adjetivos

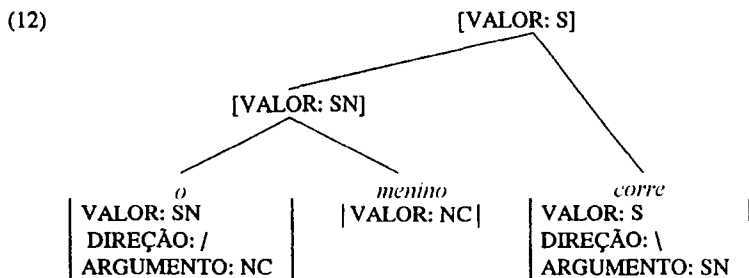
VALOR: NC	
DIREÇÃO: \	
ARGUMENTO: NC	
TRAÇOS:	GÊNERO
	NÚMERO

c. Nomes comuns

VALOR: NC	
TRAÇOS:	GÊNERO
	NÚMERO

A adição de duas categorias, que em outras versões da gramática categorial é chamada “redução”, na GUC nós chamamos de unificação de traços. O mecanismo consiste no seguinte: a operação que concatena dois elementos para formar um terceiro *unifica* os traços dos dois elementos constituintes para estabelecer os traços do elemento resultante. Em linhas gerais, o processo de unificação obedece aos seguintes princípios: 1. a unificação de duas variáveis é uma variável; 2. a unificação de uma variável e uma constante é a constante; 3. a unificação de duas constantes sempre falha (exceto se forem duas ocorrências de uma mesma constante).

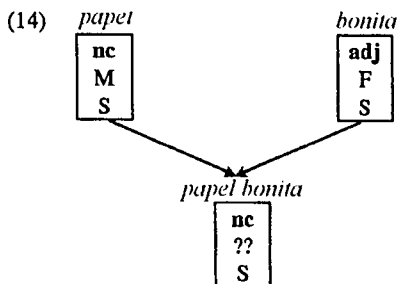
O que é resolvido, em outras versões da gramática categorial, através de regras, na GUC é resolvido através de unificação. A redução de uma expressão formada por um quantificador e um nome comum em um sintagma nominal, por exemplo, pode ser vista a seguir (estamos excluindo os traços morfológicos, já que vamos tratar especificamente deles a seguir).



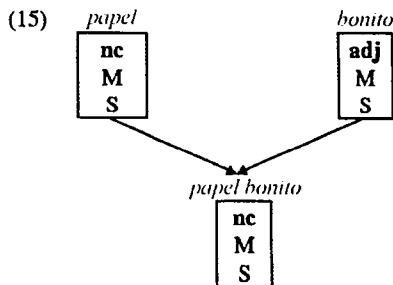
A unificação de traços só é possível quando (i) se apõe, a uma categoria complexa não-preenchida, uma categoria-argumento correta (quando se apõe um nome comum a um artigo) e (ii) quando a categoria-argumento é aposta do lado correto. Quando as duas condições são satisfeitas, obtém-se a *unificação* das duas categorias em uma categoria preenchida ou plena (quer dizer, uma categoria que só tem traço de valor).

Vejamos, agora, como funciona, dentro de uma GUC, a checagem de traços de que necessitamos para dar conta da concordância dentro do SN do português, i.e., os traços que chamamos, acima, de traços morfológicos. Por exemplo, suponhamos as três expressões seguintes: *papel verde*, *papel bonito* e *papel bonita*. Nos três casos, temos uma seqüência de *nc* e *adj*. Nos três casos, a regra *1b* (operação f_{1b}) pode se aplicar, resultando num novo *nc*. Nos três casos, no entanto, o processo de unificação vai apresentar resultados diversos. Em *papel verde*, por exemplo, o adjetivo *verde* não é marcado para gênero, ou seja, o traço de gênero é preenchido por uma variável α . Como o traço de gênero é preenchido pela constante *M* em *papel*, e a unificação de uma constante e uma variável é sempre a constante, *papel verde* vai ser marcado com o traço “M” para gênero. No caso do traço de número, como tanto *papel* quanto *verde* vêm preenchidos com a constante *S*, indicadora de *singular*, a unificação simplesmente vai reproduzir a constante na expressão resultante. Como se pode ver no exemplo (13):

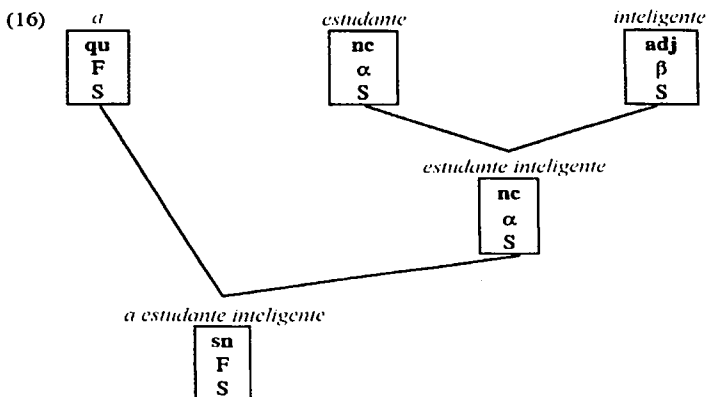
O outro caso interessante é o de *papel bonita*. Nesse caso, a unificação falha, já que *papel* tem a posição de gênero preenchida com a constante *M* e *bonita* tem a mesma posição preenchida com a constante *F*. A unificação não é possível e a expressão é considerada “agramatical”. Observe (14):



O último caso, *papel bonito*, é o caso em que a unificação vai se dar sem maiores surpresas. As duas expressões básicas que estão sendo concatenadas apresentam as mesmas constantes tanto para gênero quanto para número, o que faz com que a expressão resultante apenas repita essas constantes. Como pode se ver em (15):



Vejamos agora um *sn* um pouco mais complexo:



Nem *estudante* nem *inteligente* apresentam marcação para o traço de gênero: nos dois casos o gênero é preenchido com uma variável. Duas variáveis se unificam como uma variável. Assim, *estudante inteligente* também vai ter uma variável na posição do traço de gênero. O quantificador *a*, no entanto, vem marcado como *F* (feminino). Logo, o *sn a estudante inteligente* será marcado como *F*. É interessante observar que se o quantificador fosse masculino (marcado *M* para gênero), a unificação também seria possível e o *sn* resultante (*o estudante inteligente*, no caso) seria também masculino.

Conclusão

O mecanismo formal que exemplificamos acima, em princípio, deve dar conta dos fenômenos da concordância nominal (ao menos na maioria dos casos). Os casos “especiais” de concordância nominal (concordância dos adjetivos predicativos, concordância dos adjetivos referidos a mais de um nome comum, etc.) dependeriam de refinamentos nas regras do “parser” sintático, bem como da inclusão de novos traços nas entradas lexicais. Não foi nossa intenção apresentar de forma completa o fragmento de gramática responsável pelo tratamento da concordância; apenas nos interessava exemplificar um tipo de mecanismo que tem se mostrado útil nas tarefas inerentes à interface entre a lingüística e a computação, se o enfoque “lingüisticamente orientado” é assumido.

RESUMO

Neste texto, buscamos fazer uma apresentação, de forma clara e simples, do funcionamento de uma gramática de unificação categorial (GUC). Acrescentamos às gramáticas categoriais “clássicas” procedimentos de “unificação”, obtendo um mecanismo teórico que pode permitir um tratamento computacional adequado dos fenômenos da concordância. A exemplificação é feita com os fenômenos da concordância nominal em português.

Palavras-chave: gramática categorial, unificação, concordância, processamento de língua natural.

ABSTRACT

In this paper, we attempt to introduce a simple and straightforward application of a categorial unification grammar (CUG). We have added unification procedures to standard categorial frameworks thus obtaining a theoretical mechanism which allows for a computationally adequate treatment of agreement phenomena. Illustration from nominal agreement in Portuguese is provided.

Key-words: categorial grammar, unification, agreement, natural language processing.

REFERÊNCIAS BIBLIOGRÁFICAS

- BORGES NETO, J. Da gramática categorial. *Construtura*, Curitiba, v. 17, p. 5-19, 1979.
- BOUMA, G. Modifiers and specifiers in categorial unification grammar. *Linguistics*, v. 26, p. 21-46, 1988.
- GUIMARÃES, M. R. *Gramática categorial de unificação: uma introdução*. Comunicação apresentada no Seminário do GEL. Campinas, maio de 1997.
- LYONS, J. *Introdução à lingüística teórica*. São Paulo: Nacional/Edusp, 1979.
- UZKOREIT, H. Categorial unification grammars. In: Coling86 [1986: Bonn]. *Proceedings of Coling86*. Bonn: Association for Computational Linguistics, 1986. p. 187-194.
- WOOD, M. M. *Categorial grammars*. London: Routledge, 1993.
- ZEEVAT, H.; KLEIN, E.; CALDER, J. Unification categorial grammar. *Lingua e Stile*, v. 26, n. 4, 1991, p. 499-527, 1987.