

Recursividade na língua e habilidades cognitivas superiores

Desde o seu surgimento, nos anos 50, a Teoria Gerativa tem enfatizado o caráter recursivo da sintaxe como uma das características cruciais das línguas humanas. São freqüentes também na literatura as analogias, baseadas nessa propriedade em particular, entre o sistema numérico e a língua (cf. Chomsky, 1998; 2007, dentre outros). Entretanto, apesar do seu uso bastante difundido na literatura, o conceito de recursividade, aplicado tanto ao domínio da língua quanto a outros campos, não tem recebido uma definição clara e unívoca. Percebe-se que, até pouco tempo atrás, não havia na literatura uma preocupação manifesta por esclarecer os pontos obscuros associados à noção. Essa situação tem começado a mudar recentemente com a publicação de alguns trabalhos que visam a discutir os alcances e limites do conceito, tanto no seio da Teoria Lingüística quanto no que diz respeito a sua aplicação nas Ciências Cognitivas de um modo geral (Arsenijevic & Hinzen, 2010; Lobina & García-Albea, 2009; Tomalin, 2007).

Parker (2006a; 2006b) destaca que as definições apresentadas na Teoria Lingüística são freqüentemente “opacas”. Esse parece não ser, contudo, um problema exclusivo da lingüística uma vez que, segundo a autora, na Ciência da Computação, da qual a lingüística herdou a noção, as definições careceriam de um fio condutor comum. Também na Matemática, campo no qual o termo foi originalmente cunhado, registra-se uma situação similar (cf. Soare, 1996).

Pode-se afirmar assim que *recursividade* é um termo potencialmente problemático (Parker, 2006a e 2006b; Lobina & García-Albea, 2009; dentre outros). Nesse sentido, este capítulo tem como objetivos: explorar essa noção – tradicionalmente considerada como uma propriedade central nas línguas naturais – e discutir, em que medida e sob quais aspectos, a recursividade poderia desempenhar um papel no modo como a língua interage com outros domínios cognitivos. Cabe, pois, formular as seguintes questões: A que se refere exatamente o termo recursividade no âmbito da lingüística? Em que sentido pode-se falar de recursividade em outros domínios cognitivos fora da linguagem? Pode-se oferecer uma definição

geral de recursividade, com aplicabilidade inter-domínios?

As seguintes seções visam a iluminar as questões problemáticas acima colocadas. Em primeiro lugar, são discutidos os alcances e limites do termo *recursividade* tanto na Teoria Lingüística quanto em outros campos do conhecimento, quais sejam, a Matemática e a Ciência da Computação. Essa discussão é conduzida a partir de três pontos centrais: (a) as interpretações associadas ao termo recursividade na Matemática e na Ciência da Computação – disciplinas das quais provém o conceito incorporado pela lingüística; (b) a maneira como a noção de recursividade foi incorporada na Teoria Lingüística a partir dos anos 50 e (c) o impacto das duas questões anteriores na interpretação de recursividade no contexto do PM.

4.1

O conceito de recursividade na teoria lingüística

Apesar de tanto a Matemática quanto a Ciência da Computação compartilharem uma caracterização geral de recursividade, é possível estabelecer algumas distinções específicas para cada domínio. Em primeiro lugar, a noção de recursividade tem a sua origem na Matemática. Nessa ciência em particular, a recursividade aparece associada à idéia de um método para definir funções (e outros objetos ou relações) e é definida sempre com relação a um domínio de aplicação bem estabelecido. Pode-se dizer que a recursividade é caracterizada como uma propriedade de certos objetos, relações ou mecanismos. Encontramos então “conjuntos recursivos”, “algoritmos recursivos”, “funções recursivas”, “problemas com soluções recursivas”, “relações recursivas”, “definições recursivas”, etc.¹

Nesse âmbito, o conceito de *recursividade* aparece às vezes definido de forma bastante geral:

¹ Se dice que una función es recursiva si existe un procedimiento efectivo (algoritmo) para computarla. Se dice que un conjunto es recursivo si existe un algoritmo para decidir si un elemento pertenece o no a él. Un conjunto es recursivamente enumerable si existe un procedimiento efectivo para generar sus elementos uno después de otro (Ej. El conjunto de los cuadrados de los enteros es recursivamente enumerable...). Todo conjunto recursivo es recursivamente enumerable, pero no viceversa. (*Estructuras Sintácticas*, 1957 [versão espanhola, 1974. Nota do tradutor]).

A method of defining functions studied in the theory of algorithms and other branches of mathematical logic. This method has been used for a long time in arithmetic to define sequences of numbers (progressions, Fibonacci numbers, etc.). Recursion plays an important role in computational mathematics (recursive methods). Finally, in set theory transfinite recursion is often used. For a long time the term ‘recursion’ was used by mathematicians without being accurately defined (Vinogradov, 1992:15)

Na Computação, por sua vez, a recursividade é tomada como uma ferramenta ou uma técnica de programação. A diferença básica entre uma função recursiva definida matematicamente e uma função programada, por exemplo, a função fatorial $(n!)^2$, reside no fato de que enquanto a primeira descreve *o que é* o fatorial, a segunda descreve *como* se pode obtê-lo. Parker (2006a; 2006b) chama atenção para o fato de que na Computação as definições de recursividade carecem de um fio condutor comum e estabelece uma distinção entre as definições fornecidas. Em algumas definições a ênfase é colocada na repetição de um dado processo:

Recursion and iteration are two equivalent ways in programming for repeatedly performing a specific task (Loeper et al., 1996:153).

Recursion and iteration both result in the controlled repeated execution of a body of code (Arnou & Weiss, 1998:494).

Há também definições nas quais a *auto-referência* é considerada a característica mais marcante; ou seja, o fato de que um objeto seja definido em termos de si mesmo (*recursividade estrutural*). Por último, algumas definições se baseiam no fato de que determinados processos ou procedimentos podem se invocar a si mesmos

² O fatorial de um número natural n (representado $n!$) é o produto de todos os inteiros positivos menores ou iguais a n . A definição de fatorial pode ser formalizada da seguinte forma:

$$\begin{aligned} n! &= \text{se } n = 0 \rightarrow 1 \\ &= \text{se } n \geq 1 \rightarrow (n-1)! \end{aligned}$$

Vejamos por exemplo o cálculo de $3!$:

$$\begin{aligned} 3! &= 3 \cdot (3-1)! \\ &= 3 \cdot 2! \\ &= 3 \cdot 2 \cdot (2-1)! \\ &= 3 \cdot 2 \cdot 1! \\ &= 3 \cdot 2 \cdot 1 \cdot (1-1)! \\ &= 3 \cdot 2 \cdot 1 \cdot 0! \\ &= 3 \cdot 2 \cdot 1 \cdot 1 \\ &= 6 \end{aligned}$$

– colocado em termos computacionais, um algoritmo ou método pode se “chamar a si mesmo” (*recursividade procedimental*).

Recursion allows something to be defined in terms of smaller instances of itself (Loudon, 1999).

Recursion is the ability of a subroutine or procedure to call itself (Harel, 1993:31).

Segundo Odifreddi (2005), a *iteração* constitui o tipo mais simples possível de recursividade. Na Computação, no entanto, esses dois conceitos têm diferenças importantes. *Iteração* é definida como a repetição de instruções em um programa e a noção é utilizada em dois sentidos: (a) como termo genérico, sinônimo de “repetição”; (b) como um meio de descrever uma forma específica de repetição com um estado mutável. Conforme a primeira acepção, a recursividade seria um exemplo de iteração, mas que faz uso da sua própria notação. Já no segundo sentido, *iteração* descreve um estilo de programação utilizado em linguagens de programação imperativa que se contrapõe à recursividade que tem um enfoque mais declarativo³.

A noção de recursividade adotada pela lingüística é herança da Ciência da Computação. O próprio conceito de *computação* também constitui uma noção de grande relevância no desenvolvimento da Teoria Lingüística. *Computar* implica obter uma solução ou resultado a partir de certos dados ou entradas (números ou símbolos) utilizando para isto um procedimento explícito ou algoritmo.

Na década de 30, Alan Turing, forneceu uma formalização dos conceitos de

³ Uma das possíveis formas de classificar linguagens de programação é feita com base no paradigma que utilizam. Um paradigma em programação fornece (e determina) a visão e os métodos do programador na construção de um programa. Diferentes paradigmas resultam em diferentes estilos e formas de se pensar a solução de problemas. Existem vários paradigmas e há conflitos nas definições e alcances de alguns deles. O *paradigma imperativo*, anteriormente associado à *iteração*, descreve a programação como uma seqüência de instruções ou comandos que alteram o estado de um programa. O *paradigma declarativo*, não se baseia no *como é feito algo* (como lograr um objetivo passo a passo), mas descreve (*declara*) *como é algo*; i.e. objetiva descrever as propriedades da solução buscada, deixando indeterminado o algoritmo (conjunto de instruções) utilizado para encontrar a solução. Esse paradigma apresenta as desvantagens de ser mais difícil de implementar e menos eficiente que o imperativo, mas é particularmente vantajoso na resolução de certos problemas. A escolha da forma recursiva ou iterativa para a resolução de um problema depende de vários fatores, quais sejam: (i) a carga computacional; (ii) a redundância: algumas soluções recursivas resolvem o mesmo problema em reiteradas ocasiões; (iii) a complexidade da solução: em algumas situações é muito difícil achar uma solução iterativa; (iv) a concisão, legibilidade e elegância do código resultante.

algoritmo e computação. No seu trabalho *On computable numbers, with an application to the Entscheidungsproblem*, Turing (1936) estudou a questão, formulada por Hilbert, de enunciados matemáticos serem ou não decidíveis. Em outras palavras, se há um método definido ou algoritmo que aplicado a qualquer sentença matemática possa determinar o valor de verdade da mesma (Problema da Decisão). Turing (1936) postulou uma máquina de computar universal (*universal computer machine*), conhecida popularmente como “máquina de Turing”. Os resultados obtidos a partir dessa máquina tiveram várias aplicações, dentre as quais, uma foi a prova da insolubilidade do *Entscheidungsproblem*.

Na Matemática e nas Ciências da Computação, um algoritmo é uma lista bem definida, ordenada e finita de operações que permitem obter um resultado. Dado um estado inicial e uma entrada (*input*), através de passos sucessivos e claramente definidos, se chega a um estado final obtendo-se um *output*. Algoritmos podem ser expressos de muitas maneiras dentre as quais: linguagem natural, diagramas de fluxo, linguagens de computação, etc. O modelo computacional de Turing permitiu formalizar a definição de algoritmo, provendo ainda uma definição de computação independente do formalismo utilizado para implementá-la. Esse último ponto foi fundamental na constituição das Ciências Cognitivas já que permitiu a caracterização de uma computação simbólica, definida independentemente de um tipo particular de formalismo. A computação simbólica incorpora o conceito de “símbolo” que, nesse caso, pode ser definido em termos de uma entidade abstrata desprovida de sentido que funciona como *input* inicial em um processo ou que é o produto de sucessivas recodificações do *input* inicial. O símbolo como produto de recodificação é equivalente a uma representação dos dados de origem (Corrêa, em preparação).

Procedimentos algorítmicos mostram-se adequados para caracterizar processos mentais que transcorrem de forma regular, determinística e fora do controle do indivíduo, sendo o processamento sintático de enunciados lingüísticos um bom exemplo destes. A Psicolingüística tem se dedicado a caracterizar procedimentos de natureza algorítmica vinculados à análise (*parsing*) e formulação de enunciados verbais conduzidas em tempo real. Um paralelo pode ser estabelecido entre tais procedimentos e aquilo que é apresentado na Teoria Gerativa em termos de regras ou

operações sintáticas e, nesse sentido, pode-se afirmar que tanto a Lingüística quanto a Psicolingüística incorporam a concepção de computação simbólica na sua teorização. É importante lembrar, contudo, que a Lingüística visa a prover uma caracterização o mais abstrata possível das relações estruturais entre os constituintes de uma sentença, independentemente das condições de produção e compreensão. A Psicolingüística, por sua vez, tenta caracterizar a computação em tempo real, seguindo algoritmos com especificidades em função das condições de produção e de compreensão.

A concepção de gramática em termos algorítmicos representou uma alteração fundamental no pensamento lingüístico a partir da segunda metade do século XX. Até então, a lingüística estruturalista, na sua vertente norte-americana, havia realizado uma descrição das relações estruturais em termos de constituintes imediatos. Chomsky adota a idéia de uma gramática sintagmática (inicialmente introduzida por Zellig Harris), definida como um procedimento algorítmico que partindo de um símbolo inicial (S, a maior unidade de descrição sintática) deriva sentenças da língua. As etapas de derivação de uma sentença conforme esse tipo de algoritmo, são implementadas mediante regras de reescritura que explicitam relações de dependência entre os constituintes. A reintrodução de um símbolo como produto da reescritura fornece à gramática um caráter recursivo e garante que um número infinito de sentenças seja gerado por meio de um número finito de regras. Em suma, pode-se dizer que a concepção de gramática em termos algorítmicos – ou seja, a concepção de uma gramática gerativa que tomou forma em meados do século XX – é função de desenvolvimentos do âmbito da Matemática e da Ciência da Computação⁴.

A noção de *recursividade* tem tido desde sempre um lugar de destaque na Teoria Gerativa na hora de fornecer uma explicação para algumas das características mais salientes das línguas humanas, em particular, a *infinitude discreta* (i.e. a possibilidade de que um número potencialmente infinito de novas sentenças seja

⁴ A proposta de Turing não foi, contudo, a única influência decisiva na conformação da Teoria Gerativa. Tomalin (2007) salienta o impacto que as idéias de Bar-Hillel tiveram no desenvolvimento da teoria chomskyana, cuja importância foi reconhecida pelo próprio Chomsky (cf. Chomsky, 1955:45). Bar-Hillel (1953) argumentou que definições recursivas não ficavam restritas apenas ao contexto da matemática pura e da lógica, mas que esse tipo de definição poderia vir a facilitar a metodologia analítica adotada por ciências empíricas como a lingüística.

gerado a partir de um conjunto discreto de elementos⁵). Embora o termo seja comumente associado à noção de *produtividade* ou *criatividade* lingüística, quando falamos em recursividade, essa noção arrola um conjunto maior de propriedades. Originalmente, a Teoria Lingüística parece ter adotado a noção de recursividade como uma ferramenta útil para reduzir a complexidade da gramática (Ćavar & Ćavar, 2007), por um lado e como meio de enfatizar a sua capacidade produtiva, por outro:

If a grammar has no recursive steps [...] it will be prohibitively complex [...]. If it does have recursive devices, it will produce infinitely many sentences. (Chomsky, 1956:116).

A formulação de dois tipos de mecanismos recursivos pode ser identificada num dos trabalhos mais antigos de Chomsky (*The Logical Structure of Linguistic Theory*, 1975 [1955]). O primeiro envolve a aplicação sucessiva de regras da gramática seqüencialmente ordenadas, enquanto que o segundo deriva da inclusão de regras na gramática que são elas mesmas recursivas por definição (Tomalin, 2007).

Nas primeiras versões da teoria, a recursividade lingüística ficava definida de um modo bastante simples a partir das regras de reescritura ($S \rightarrow NP VP \dots$). Posteriormente, a partir de Chomsky (1965), a recursividade passa a ser codificada na *Estrutura Profunda* (*Deep Structure*- DS) (cf. Hornstein et al., 2005). Atualmente, apenas regras de seleção são postuladas e DS foi eliminado dos níveis de representação. Ao invés desses recursos formais, é assumida uma operação que combina os itens lexicais e os organiza numa estrutura frasal⁶. Essa operação é chamada de *Merge*. Pode-se considerar, contudo, que a mudança principal com relação às regras diz respeito ao formalismo utilizado. Nesse sentido, quando se diz que os traços de um núcleo são projetados quando este se combina (*merge*) com outro elemento, tem-se uma expressão equivalente à re-introdução de um símbolo numa regra de re-escritura. Cabe lembrar que a direcionalidade da derivação foi também afetada com essa mudança, passando de *top-down* a *bottom-up*.

Embora as regras de re-escritura tenham sido dispensadas do atual quadro teórico, alguns autores continuam definindo recursividade nesses termos:

⁵ *O uso infinito de meios finitos* em palavras de Humboldt.

⁶ Essa estrutura segue a teoria-X' nas versões mais antigas. Propostas mais recentes assumem a *bare phrase structure*, embora nem todos os autores usem esse tipo de notação.

[...] recursion implies that a grammatical symbol could be rewritten by using a symbol of the same category. In more technical terms a symbol can occur both on the left and on the right side of the rewriting rule. In this less restrictive perspective, recursion derives directly by the application of the simplest combinatorial operation of (morpho-) syntax, i.e. “merge” in the sense proposed by Chomsky 1995: if A and B combine and either A or B projects, we derive a category which contains itself as a part (Tommaselli, 2005: 111- 112).

Parker (2006a) delimita quatro grupos de caracterizações para o conceito de recursividade. O critério para essa divisão é estabelecido com base no aspecto que resulta mais saliente em cada definição. Conforme essa classificação, na Teoria Lingüística o termo em questão é utilizado como sendo quase um sinônimo de:

- (i) *Infinitude discreta* (Adger, 2003; Lobeck, 2000; Carnie, 2002);
- (b) *Estrutura frasal* (Lobeck, 2000; Pinker, 2003; Horrocks, 1987);
- (c) *Encaixamento*, especialmente de constituintes da mesma natureza (Kirby, 2002; Pinker & Jackendoff, 2005); e
- (d) *Iteração* (Radford, 1997; Hurford, 2004; Pinker & Jackendoff, 2005).

Pinker e Jackendoff (2005) por exemplo, fornecem uma definição computacional de recursividade baseada na estrutura frasal:

Recursion refers to a procedure that calls itself, or to a constituent that contains a constituent of the same kind (Pinker & Jackendoff, 2005:3).

As definições por vezes combinam vários dos aspectos mencionados, especialmente estrutura frasal e iteração:

From the point of view of linguistic theory, the concept of recursion could be defined in both a narrow sense and in a broader sense. In the narrow sense, recursion implies subordination.

- (i) I think/hope/promise that John will come tomorrow
 $S \rightarrow [... V S]$

In a broad sense, recursion as a computational system could be defined in logical terms as follows:

- (ii) $X \rightarrow [...X...]$.

(Chomsky 1965:211)

Outras definições conjugam todos os aspectos anteriormente citados:

La ricorsività sintattica significa che una determinata

categoria compare come elemento all'interno di un sintagma appartenente a questa stessa categoria. L'esempio più tipico di questo fenomeno sono le frasi che compaiono all'interno di altre frasi. [...] La ricorsività sintattica significa che le stesse regole sintattiche possono venir riapplicate ciclicamente all'interno di una stessa frase. In questo modo si riesce a chiarire come la sintassi del linguaggio umano sia uno dei principali fattori che spiegano il carattere aperto di questo particolare sistema di comunicazione, cioè la sua capacità di generare frasi sempre nuove (Parisi, 1981:25-26).

Chomsky (2000), nas versões mais recentes da teoria, utiliza o termo recursividade como sendo sinônimo de infinitude discreta, que, por sua vez, é tomada muitas vezes como sinônimo de criatividade e produtividade lingüística. Cabe lembrar que é precisamente com base na propriedade da infinitude discreta que mais freqüentemente é estabelecido um paralelo entre língua e o sistema dos números naturais:

Human language is based on an elementary property that also seems to be biologically isolated: the property of discrete infinity, which is exhibited in its purest form by the system of natural numbers 1, 2, 3... Children do not learn this property of the number system. Unless the mind already possesses the basic principles, no amount of evidence could provide them, and they are completely beyond the intellectual range of other organisms. (Chomsky, 2000: 3-4)

Apesar do uso por vezes intercambiável entre algumas das noções discutidas, é possível estabelecer algumas distinções relevantes. A primeira diz respeito à diferença entre recursividade e iteração. Sob um ponto de vista amplo, a iteração aparece como sendo o tipo mais simples de recursividade. No que diz respeito à língua, contudo, pode ser estabelecida uma distinção entre ambos os processos com base na diferença entre encaixamento e repetição. Assim, enquanto a iteração implica a repetição de uma ação ou de um objeto um número arbitrário de vezes, a recursividade envolve o encaixamento de uma ação ou objeto em outra instância de si mesmo⁷.

⁷ *Iteration*: the simple un-embedded repetition of an action or object an arbitrary number of times.

Recursion: the embedding at the edge or in the center of an action or object one of the same

Uma segunda distinção pode ser estabelecida entre recursividade e estrutura frasal. Frequentemente, ambos os conceitos são tomados como sendo equivalentes, embora a idéia de estrutura frasal não precise necessariamente estar vinculada a regras recursivas (Lobina & García-Albea, 2009). Mais especificamente, a estrutura frasal se refere à relação hierárquica dos constituintes na sentença, ao passo que uma estrutura pode ser hierarquicamente organizada sem por isso ser, necessariamente, recursiva. Parker (2006b) considera que a diferença crucial radica no fato de que a hierarquia implica o encaixamento de constituintes em outros constituintes, enquanto que a recursividade envolve o encaixamento de constituintes *idênticos*, uns nos outros (um CP dentro de um CP, um DP dentro de um PP que por sua vez está incluído em outro DP, etc.).

Por último, há na literatura uma distinção com relação às características específicas do encaixamento numa dada estrutura recursiva. Essa distinção é estabelecida em termos de recursividade na borda (*tail*) ou no centro (*nested*).

The two types of recursion in question are known as **tail recursion and embedded/nested recursion**. The former is understood simply as the type of recursion which occurs at either end of a phrase or sentence. In other words, the self-embedding of phrases is at the edge. The latter type of recursion occurs in the middle of a phrase or sentence (...) **Tail recursion** is the type of recursion that exists when a phrase is embedded within a phrase of the same type only at the beginning or end of a sentence or phrase. In other words, the embedding happens at the ‘tail’ of the sentence or phrase (...) **Nested recursion** is so called because it refers to embedding that is nested inside a phrase or sentence, such that material exists on both sides of the embedding. It does not occur at either edge of the sentence. Examples of nested recursion are more difficult to construct, but centre-embeddings (...) illustrate the phenomenon. (Parker, 2006b:172).

Essa distinção pode ser ilustrada pelos seguintes três tipos de estrutura. A construção possessiva em (21) e a cláusula relativa em (22) seriam exemplos do primeiro tipo de recursividade. Já a construção com encaixamento no centro em (23) ilustraria a *nested recursion*.

type. Further, nested recursion leads to long-distance dependencies and the need to keep track, or add to memory (Parker, 2006b).

- (21) O vestido da amiga da irmã da Ana é elegante.
 (22) Este é o jogador que chutou a bola que o adversário perdeu no instante que o juiz apitou.
 (23) A menina que o menino que o professor reclamou beijou está gripada.

Enquanto a *tail recursion* envolve encaixamento na borda (*edge*) do sintagma, a *nested recursion* acarreta o encaixamento no centro, deixando material em ambos os lados do constituinte encaixado. Esse último tipo produz dependências de longa-distância. Com base nessa caracterização seria possível concluir que a *tail recursion* seria apenas um caso de iteração. Contudo, se considerarmos o exemplo em (24) observamos que essa sentença não pode ser analisada como uma proposição simples com outro NP ou DP acrescentado no começo.

- (24) A namorada do João gosta muito dele.

Assim, a interpretação do material à direita do DP recursivo (DP>PP>DP) depende crucialmente do processamento da sentença como um todo. Nesse sentido, sentenças que apresentam *tail recursion* também podem produzir relações de longa distância, embora possa ser estabelecida uma distinção entre esse tipo de dependência (no qual há material interveniente a ser integrado composicionalmente) e dependências em que há a presença de uma variável a ser imediatamente identificada (como ocorre no caso das relativas ramificadas à direita).

Além das distinções já mencionadas, outras classificações têm sido levantadas. Roeper & Snyder (2005) propõem que haveria certas formas de recursividade sujeitas a variação inter-lingüística. Fazendo um uso bastante amplo do termo, os autores sugerem que haveria, no mínimo, três tipos de recursividade: *self-embedding*, iterativa e *scopal*. O primeiro tipo se refere aos casos nos quais um constituinte estrutural é encaixado dentro de um outro constituinte maior do mesmo tipo.

A recursividade iterativa, por sua vez, inclui:

- (i) Casos nos quais um item lexical é repetido para dar ênfase, por exemplo:
very, very, [...] happy;
 (ii) Seqüências de cláusulas relativas como em *This is the cat [that ate the rat*

[*that ran out ...*]; e

(iii) Coordenação de um número arbitrário de sentenças.

As estruturas apontadas em (i), (ii) e (iii) se caracterizam, segundo os autores, pelo fato de que as operações gramaticais relevantes podem ser expressas por regras recursivas, mas o resultado corresponde à chamada *tail recursion* – i.e. um tipo de recursividade na qual a simples iteração funciona como um substituto computacional suficiente.

O último tipo de recursividade considerado pelos autores é chamado de *scopal recursion* que envolveria os casos de variáveis ligadas e polaridade negativa, normalmente descritas em termos de relações de c-comando. Essa relação é, para os autores, formalmente equivalente à propagação de informação *downward*, recursivamente, através da árvore (Ex. o escopo da negação, sentenças com verbos factivos, etc.).

Para Roeper & Snyder (2005) a recursividade teria um papel crucial na aquisição da língua, já que informaria à criança quando uma operação gramatical produtiva é aplicada. A hipótese defendida pelos autores é que o inventário de operações recursivas seria o único ponto de parametrização dentro da FLN e que a evidência de aplicação recursiva dessas operações guiaria a aquisição da sintaxe pelas crianças (Roeper & Snyder, 2004). Essa proposta parece difícil de ser diretamente compatibilizada com o modo como operações recursivas são caracterizadas, tanto em um modelo padrão quanto na proposta minimalista. Segundo a lógica minimalista, qualquer possível distinção entre tipos de recursividade estaria representada em propriedades de núcleos funcionais no léxico. Assim sendo, a possibilidade de recursividade estaria em FLN e não no modo como cada língua opta por expressá-la, o que dificultaria a implementação da idéia de parametrização de FLN tal como formulada pelos autores.

Ainda no que diz respeito ao papel da recursividade na aquisição, Roeper e colaboradores (2009; Hollebrandse & Roeper, submetido; Roeper & Snyder, 2004), estabelecem uma distinção entre recursividade *direta* (a) e *indireta* (b):

(a) Recursividade direta: $AP \Rightarrow A (e AP)$,

sendo que a regra de conjunção se aplica a qualquer categoria: N, V, P ou S (CP).

(b) Recursividade indireta: NP => N (PP)

PP => P (NP)

DP => (Poss)NP

Poss => DP 's

A recursividade “direta” teria lugar quando um elemento é concatenado com outro elemento da mesma categoria. Já a recursividade “indireta” cria um *loop* que pode ser de dois tipos:

(25) Recursividade indireta não-relevante: *the man in the middle*

[DP1 [NP [PP [DP2 NP]]]]

(26) Recursividade indireta (sujeita a restrições): *John's father's friend's hat*:

[DP [NP [DP [NP D [NP D [NP2

fase 2 fase 1

Conforme Boeckx (2008), a interpretação de uma derivação ocorre em uma sequência alternada: *Spellout* “*transfers to interpretation the complement of a phase only*”. Esse fenômeno, segundo o qual o núcleo da fase e o complemento da fase devem alternar, é chamado por Roeper (2009, dentre outros) de *Phase Alternation Constraint*. Roeper e colaboradores (2009; Hollebrandse & Roeper, submetido) consideram que haveria um contraste básico entre recursividade direta – que não requer uma interpretação periódica – e indireta que estaria submetida a restrições decorrentes da alternância de fases. Não fica claro, contudo, porque apenas estruturas com recursividade indireta estariam sujeitas a tais restrições, já que a alternância de fases nos termos propostos por Boeckx ocorreria em qualquer derivação. A diferença relevante para os autores pode residir no fato de que nas estruturas que envolvem a denominada recursividade indireta a alternância de fases ficaria “explícita” (na medida que cada constituinte na estrutura recursiva pode ser enviado para *Spellout* ainda no decorrer da derivação).

De um modo geral, e a despeito das divergências apontadas, pode-se dizer que

não há de fato uma incompatibilidade entre as diferentes definições propostas. A aparente diversidade radica basicamente no aspecto salientado em cada caso: a recursividade como fenômeno lingüístico – vinculada portanto a características específicas das línguas humanas – ou como mecanismo ou recurso computacional, abrangendo assim outros possíveis domínios.

Até aqui a noção de recursividade foi vinculada ao domínio da língua natural ou, no caso da Matemática e das Ciências da Computação, a linguagens formais que partilham propriedades com aquela. A seguir faremos referência brevemente a questão da relação entre recursividade e cognição não-lingüística. Em outras palavras, partindo do fato de que a língua é considerada um sistema recursivo, haveria outros sistemas cognitivos partilhando essa mesma propriedade? Existiria uma relação entre domínios em virtude da recursividade, definida em termos de um recurso ou mecanismo computacional mais geral?

4.2

Recursividade e cognição não lingüística

Hauser et al. (2002, 2005) consideram que a recursividade poderia ser uma propriedade presente em outros domínios da cognição, e ainda, ser compartilhada por outras espécies. São apontados vários espaços nos quais essa propriedade poderia estar presente (Corballis, 2007a). A cognição matemática, ou melhor dito, o número, é um dos candidatos plausíveis. Não é claro, contudo, se linguagem e capacidade de lidar com número são habilidades que evoluíram de maneira interligada (Chomsky, 1998; Hurford, 1987). O domínio da visão, responsável pela decomposição de elementos e cenas complexas por sua vez, também parece atuar de forma recursiva (um tipo de recursividade procedural).

A música, considerada como um tipo de linguagem especial, possui a característica de ser hierarquicamente organizada. Não é trivial, contudo, estabelecer se uma sequência musical consiste de “constituintes” repetidos que podem ser analisados de forma plenamente recursiva ou apenas iterativa. Na linguagem natural, a semântica fornece certa informação sobre a estrutura, da qual a música carece. Pode-se dizer, contudo, que a recursividade neste caso estaria na aplicação de regras

que combinam um número finito de tons de forma a dar origem a um número infinito de sentenças musicais. O tipo de regras e de restrições a estas seria específico do domínio, mas o procedimento recursivo poderia ser compartilhado. Num trabalho recente, Pesetsky & Katz (submetido) apresentaram a denominada “tese da identidade para a música e a linguagem”. Segundo os autores, todas as diferenças formais entre ambos os sistemas derivam das diferenças entre seus blocos de construção fundamentais: sons e significados arbitrariamente pareados no caso da língua natural e classes de tons e classes de combinações de tons, no caso da música. Nesta perspectiva, música e língua seriam idênticas sob qualquer outro aspecto. Especificamente, os autores afirmam que, assim como a língua, a música: *“contains a syntactic component in which headed structures are built by iterated, recursive, binary Merge”*.

Por último, a cognição espacial – ligada à navegação e localização – é outro dos domínios que tem sido relacionado à recursividade (Bartlet & Kazakov, 2004; Kazakov & Bartlet, 2005; Arsenijevic, 2008). Arsenijevic (2008) salienta uma série de semelhanças entre a cognição espacial e a língua, dentre as quais o tipo de computações presentes em cada domínio: recursiva e categorial. A cognição espacial faz parte dos *core systems* propostos por Spelke e seu funcionamento parece ser em grande medida compartilhado por outras espécies.

No que diz respeito à cognição social, a ToM – definida como a capacidade que permite representar os conhecimentos, pensamentos e sentimentos próprios e dos outros – merece um destaque especial por ser um domínio no qual a recursividade parece ter um papel crucial. Tem sido argumentado (de Villiers & de Villiers, 2000) que o desenvolvimento da ToM só é possível com ajuda de uma estrutura conceptual complexa capaz de gerar proposições recursivas. Embora a ToM envolva também habilidades relativamente mais simples, como inferir estados de ânimo a partir de expressões faciais, o foco da pesquisa tem se centrado particularmente na compreensão das chamadas crenças falsas. Essa capacidade mais sofisticada envolve o entendimento de informação diferente e variável sobre relações no mundo e implica compreender que as crenças podem variar, que ocasionalmente podem ser falsas e que com frequência são modificadas ou atualizadas com base em nova evidência

disponível. Leekam (1990) tem sugerido que a atribuição de crenças falsas envolve a representação recursiva de estados mentais.

Mas afinal, como a crença sobre a crença dos outros é representada? J. de Villiers (2007, 2006, 2004, 2003, dentre outros) assume que essa representação tem a propriedade de uma sentença recursiva – uma sentença encaixada na outra – na qual os valores de verdade de cada sentença podem ser independentes: em *Pedro acha que viu um unicórnio no jardim*, a sentença encaixada pode ser falsa enquanto a sentença como um todo permanece verdadeira. Esse tipo de estrutura estaria associado a verbos de estado mental e de comunicação e esse fato seria central para seu aprendizado. Nessa perspectiva, considera-se que apenas quando a criança tenha adquirido essa estrutura recursiva específica será capaz de raciocinar em função de crenças falsas. Verbos de comunicação e de estados mentais compartilham uma estrutura sintático-semântica recursiva crucial, e identificá-los no discurso desencadeia a compreensão e a geração de asserções acerca de crenças falsas. A aquisição das propriedades sintáticas e semânticas dos complementos sentenciais (“visíveis” nos verbos de estado mental e verbos de comunicação) seria crucial para o desenvolvimento de uma ToM representacional (de Villiers & Pyers, 2002; de Villiers, 2005; de Villiers & de Villiers, 2003; de Villiers et al., 2003; Hale & Tager-Flusberg, 2003; Lohmann & Tomassello, 2003; Dias, 1993; Dias et al., 1994, dentre outros). Entretanto, nesta perspectiva não é claro se:

- a) A recursividade estaria presente previamente no domínio da cognição social *per se* – por exemplo no nível do pensamento proposicional – e as estruturas fornecidas pela língua seriam apenas um veículo para articular certas relações de um modo mais explícito ou se, pelo contrário (Villarinho, em preparação);
- b) A recursividade lingüística alavancaria o desenvolvimento da ToM possibilitando a configuração de um pensamento proposicional recursivo.

A hipótese de que a recursividade como recurso computacional subjaz a vários outros domínios, além do lingüístico, parece muito plausível. O fato de a língua permear em boa medida a maior parte dos processos mentais na cognição adulta pode, entretanto, dificultar o acesso a dados mais conclusivos. Estudos conduzidos com outras espécies podem ser a chave para esclarecer essa questão.

Atualmente, há controvérsia quanto ao fato de a recursividade como recurso computacional ser ou não uma propriedade caracteristicamente humana. Embora haja evidências de que várias espécies são tão sensíveis à propriedade de *numerosidade* quanto bebês humanos, os animais em geral careceriam da compreensão da função de sucessor, que encontra-se na base da recursividade numérica. Nesse sentido, o número não seria um domínio suscetível de abrigar a propriedade de recursividade em outras espécies.

Resultados de estudos sobre a navegação baseados no paradigma *travelling salesman*⁸ com macacos e abelhas sugerem que esse domínio seria um bom local para procurar evidências de recursividade na cognição de outras espécies (Cramer & Gallistel, 1998; Lihoreau et al., 2010; dentre outros). Animais são capazes de computar estruturas complexas vinculadas à representação espacial do contexto (por exemplo, no que diz respeito à delimitação das rotas mais relevantes).

Seqüências de ações complexas realizadas por certos animais, tais como as encontradas nas técnicas de preparação de alimentos dos gorilas de montanha (Byrne & Russon, 1998), podem ser indicativas de raciocínio hierárquico, providenciando um campo para a testagem experimental da presença de recursividade. Resultados que atestam a capacidade de imitação de ações com estrutura hierárquica complexa em chimpanzés (Whiten, 2002) seguem a mesma direção.

A atribuição de uma ToM a outras espécies é um assunto bastante discutido. Enquanto algumas pesquisas sugerem que os chimpanzés não são capazes de representar o conteúdo da mente de outro indivíduo (Tomasello & Hare, 2003), outras indicam que ainda possuindo apenas aspectos rudimentares de ToM, outras espécies poderiam ser capazes de manipulação conceptual recursiva. Um exemplo disso seriam os babuínos que classificam a si mesmos e aos seus pares ao longo de uma linha hierárquica de dominância em grupos de tipo matrilinear. Aparentemente, esses animais são capazes de formar certas representações conceptuais estruturadas do tipo [X é mãe de Y [que é mãe de Z [que é minha mãe]]] ou [X é mais dominante

⁸ O *travelling salesman* é um problema em otimização discreta ou combinatória e pode ser formulado da seguinte forma: Dado um número de cidades e o custo da viagem de uma cidade para a outra, qual é o caminho mais econômico que pode ser traçado de modo a visitar cada cidade e depois voltar à cidade inicial?

que Y [que é mais dominante que Z [que é mais dominante que eu]]] nas quais haveria *tail recursion* associada. (Bergman et al., 2003). Uller et al. (2004) por sua vez, trazem evidências compatíveis com a idéia de que chimpanzés bebês compreendem comportamentos orientados por metas, sendo que essa compreensão denota habilidades de *mind reading* fundamentais para o desenvolvimento de uma ToM.

Os sistemas de comunicação animal são outro espaço no qual têm sido procurados sinais de recursividade. Esses sistemas podem ser divididos em dois grupos: i) aqueles com semântica delimitada e uma organização linear não-hierárquica (como a dança das abelhas ou os gritos de alarme dos macacos Campbell) (Zuberbühler, 2002); ii) aqueles com uma organização hierárquica complexa, mas sem semântica (como o canto dos pássaros) (Okanoya, 2002). Dado que carece de organização hierárquica, o primeiro grupo de linguagens pode ser descartado. Já o segundo grupo apresenta as características consideradas como necessárias para a presença de recursividade. O problema, neste caso, é como distinguir iteração de recursividade. Trabalhos recentes (Corballis, 2007b) afirmam que não existe recursividade plena no canto dos pássaros.

Até então não se têm achado evidências de *nested recursion* em outras espécies (Corballis, 2007b; Parker, 2006a, 2006b). Esse fato permite levantar a hipótese de que uma diferença crucial entre cognição humana e não-humana poderia radicar em diferenças na capacidade e no tipo de memória. Estudos recentes (Fitch & Hauser, 2004) sugerem que os macacos sagüi (*tamarins*) seriam capazes de aprender apenas cadeias com a forma $a^n b^n$, diferentemente dos humanos que também podem aprender as formas $(ab)^n$. Esses resultados podem indicar que o ponto realmente crucial na evolução da linguagem humana não teria sido a propriedade de recursividade *per se*, mas a melhora na memória do tipo “pilha” (*stack-type memory*) necessária especificamente para lidar com *nested recursion*. Trabalhos com primatas não humanos (Read, 2006) sugerem que diferenças na memória de trabalho podem explicar a falta de evidência de recursividade na cognição de outras espécies. Assim, a capacidade da memória de trabalho poderia ser o limite cognitivo que define a presença ou a ausência de pensamento recursivamente organizado.

Em suma, ainda não há evidências conclusivas que apontem para a presença de recursividade do tipo *nested* em animais não humanos. Nesse sentido, a questão da recursividade – tal e como caracterizada quando associada à linguagem humana – ser ou não uma propriedade específica da nossa espécie, continua em aberto.

Todavia, especula-se que efetivamente, outros domínios da cognição humana possam apresentar essa propriedade. Além da cognição numérica, a ToM parece ser o domínio em que a recursividade teria um papel mais evidente, resta saber ainda qual a real importância da língua nesse caso (Marcilese et al., 2009; Villarinho & Marcilese, 2009).

4.3

Recursividade nesta pesquisa

No atual *framework* minimalista, o conceito de *numeração* tem ganhado uma maior relevância. É importante lembrar que a numeração não constitui um nível sintático de representação – já que não contém primitivos nem operações específicas – mas consiste num conjunto não estruturado⁹ de pares de itens do tipo (l, n) , sendo que l é uma unidade do léxico e n um índice que indica o número de vezes que o item l foi selecionado. Assim sendo, o limite da aplicação da operação *Merge* durante a derivação de uma dada sentença fica restrito ao conteúdo da numeração, já que a derivação sintática termina quando todos os índices da mesma são exauridos. Ainda que, idealmente, o sistema computacional possa gerar sentenças de tamanho infinitamente grande, limites decorrentes de processamento fazem com que tais sentenças não sejam espontaneamente produzidas pelos falantes. Essa possibilidade está, não obstante, virtualmente presente na língua.

Crucialmente assumimos aqui que não haveria *tipos de recursividade*, mas diferentes estruturas sintáticas recursivas. Assim, as diferenças entre essas estruturas

⁹ Embora inicialmente a numeração seja concebida como um conjunto não estruturado, a ideia de *sub-arranjos* na numeração, associada ao conceito de *fase* envolve certa organização do conjunto.

Na perspectiva de um modelo de processamento – tal como o assumido nesta pesquisa –, certamente a proposição (originada na intenção de fala do falante) que determina a seleção dos itens do léxico que compõem uma dada numeração, deve ter um papel crucial na conformação de possíveis sub-arranjos.

não são derivadas da recursividade em si, mas das propriedades dos itens do léxico e das projeções envolvidas. Assume-se assim que para haver recursividade na língua é necessário que haja:

- (i) Uma estrutura hierárquica que exprima a propriedade de inclusão, “ser parte de”; e
- (ii) Uma relação de identidade (X é parte do próprio X).

Conforme essa definição, diversos tipos de estrutura podem ser caracterizados como recursivos, embora seja possível estabelecer uma distinção entre essas estruturas em termos de custo e complexidade computacional. Um exemplo disso são as construções – tradicionalmente caracterizadas como recursivas – que apresentam um uso reiterado do núcleo C (complementizador) e que admitem configurações envolvendo relações de complementação ou de adjunção no caso das sentenças relativas e adverbiais. O custo de processamento desses tipos de estruturas é diferenciado, mas a recursividade em si é a mesma.

Nesta pesquisa, a recursividade na língua fica definida pela possibilidade de que um nó funcional possa ser selecionado *n* vezes de um arranjo (numeração) para entrar em uma mesma estrutura de c-comando durante uma derivação lingüística. Isso é possível na medida em que as operações do sistema computacional podem ser aplicadas reiteradamente, tendo como único limite o número de elementos contidos na numeração. Esse modo de conceber em que consistiria estritamente a recursividade lingüística não implica que outros aspectos, freqüentemente associados ou tomados como sinônimos de recursividade, sejam desconsiderados. O objetivo é apenas delimitar claramente o que será considerado como sendo lingüisticamente recursivo no âmbito deste trabalho.

No que diz respeito à recursividade em termos de um recurso ou mecanismo computacional – isto é, não restrito apenas ao domínio da língua – a definição proposta por Pinker & Jackendoff (2005) anteriormente apresentada e repetida abaixo, parece abrangente o suficiente para cobrir um amplo espectro de fenômenos:

Recursion refers to a procedure that calls itself, or to a constituent that contains a constituent of the same kind (Pinker & Jackendoff, 2005:3).

Nota-se que a propriedade de inclusão (“ser parte de”) e a relação de identidade (X é parte do próprio X), consideradas na caracterização da recursividade lingüística, se mantêm numa definição mais geral.

Neste capítulo buscou-se oferecer um panorama dos vários desdobramentos que o conceito de recursividade apresenta, não só no que diz respeito a seu tratamento na Teoria Lingüística, mas também em outras áreas de conhecimento. Foi explicitada ainda a noção adotada especificamente nesta pesquisa. Questões vinculadas a possível relação entre recursividade lingüística e cognição numérica são abordadas experimentalmente no capítulo 6.

Os próximos dois capítulos reportam os experimentos conduzidos no âmbito desta tese. Pelo fato de termos incluído nesta pesquisa tanto crianças de várias faixas etárias quanto adultos, a metodologia utilizada foi bastante variada. As técnicas empregadas em cada experimento são detalhadas no começo das seções correspondentes.