

O problema da ordem dos modificadores adjectivais em português e em inglês no contexto de um sistema de Processamento de Linguagem Natural bilingue

ANDRÉ ELISEU

(ILTEC, Faculdade de Letras da Universidade de Lisboa)

ANA LÚCIA SANTOS

(ILTEC)

ERMELINDA GONÇALINHO

(ILTEC)

Esta comunicação pretende apresentar uma proposta para o tratamento computacional das estruturas de modificação, que exibem diferentes ordens dos modificadores adjectivais em inglês e em português, no contexto do projecto GLEP¹.

A obtenção da correcta ordem de modificadores em português a partir da ordem no inglês é um problema não trivial, dadas as restrições que caracterizam o formalismo utilizado na escrita da gramática. O problema que nos ocupa pode ser ilustrado por casos como:

- (1) a. The best electrical engine.
b. O melhor motor eléctrico

As soluções propostas para o tratamento deste problema no quadro do formalismo usado pelo GLEP² não são satisfatórias pelas razões adiante apresentadas; pelo contrário, a generalização proposta por Cinque (1996) parece dar conta adequadamente da diferente distribuição dos modificadores adjectivais em inglês e em português. No entanto, a ideia central de Cinque — a de que a diferente ordem reflecte a distribuição de adjectivos atributivos e predicativos — não é directamente implementável na linguagem usada pelo projecto. Desta forma, teve que se procurar uma solução original que permitisse expressar

a solução proposta por Cinque (1996) no formalismo adoptado (que consiste numa versão da teoria HPSG (Head-driven Phrase Structure Grammar) implementada em ALEP³).

O projecto GLEP visa desenvolver um protótipo de um sistema de tradução automática (TA) baseado na plataforma ALEP, a qual permite a expressão de descrições linguísticas segundo um formalismo baseado na teoria da HPSG, de Pollard e Sag (1987, 1994). O projecto, que reutiliza os resultados obtidos no âmbito do projecto europeu LSGram, no qual esta equipa participou, e a investigação feita no âmbito do projecto GSP⁴, tem como objectivo a construção de um sistema de tradução automática capaz de lidar com texto real, tendo o inglês como língua de partida e o português como língua alvo.

A presente comunicação é estruturada da seguinte forma: numa primeira parte será feita a caracterização do projecto no contexto dos sistemas de TA e a apresentação do tratamento formal do problema no quadro da teoria da HPSG; na segunda parte, será apresentada a proposta de tratamento para o problema da ordem dos modificadores no quadro do projecto GLEP.

Dada a arquitectura da plataforma utilizada, um sistema de TA baseado em ALEP tem necessariamente de adoptar a estratégia de 'transfer'.

Os vários modelos correntes — adoptados, em diferentes graus, por sistemas experimentais ou comerciais — distinguem-se pela estratégia proposta para lidar com a passagem de informação linguística da língua fonte para a língua alvo. Evidentemente, o tipo de operações e a natureza dos objectos linguísticos envolvidos nessa passagem dependem crucialmente da estratégia adoptada: assim, no caso dos modelos ditos de 'interlingua', que propõem uma representação comum ao par de línguas envolvido na operação de tradução, visa-se obter uma representação comum de natureza semântica (idealmente, uma representação puramente semântica). Desta forma, as operações básicas do processo de análise visam a computação dos valores semânticos considerados comuns ao par em causa⁵. No caso do modelo de 'transfer', a operação de transferência de informação é (virtualmente) uma operação análoga a qualquer outra operação desempenhada pelo sistema com vista a obter uma representação a partir de outra representação⁶.

Dito de outro modo: a tradução de uma expressão linguística em L_F numa expressão bem formada em L_A não pode ser descrita como uma operação simples $T_F \rightarrow T_A$ mas antes como uma sequência complexa de operações que, dada a monotonicidade dos sistemas de TA, de uma forma simples ("one shot derivation"), deriva representações de nível L a partir de uma expressão do nível $L-1$; uma sequência ordenada e completa de operações deste tipo constitui uma tradução. Assim, o esquema acima apresentado terá de ser desdobrado em, por exemplo,

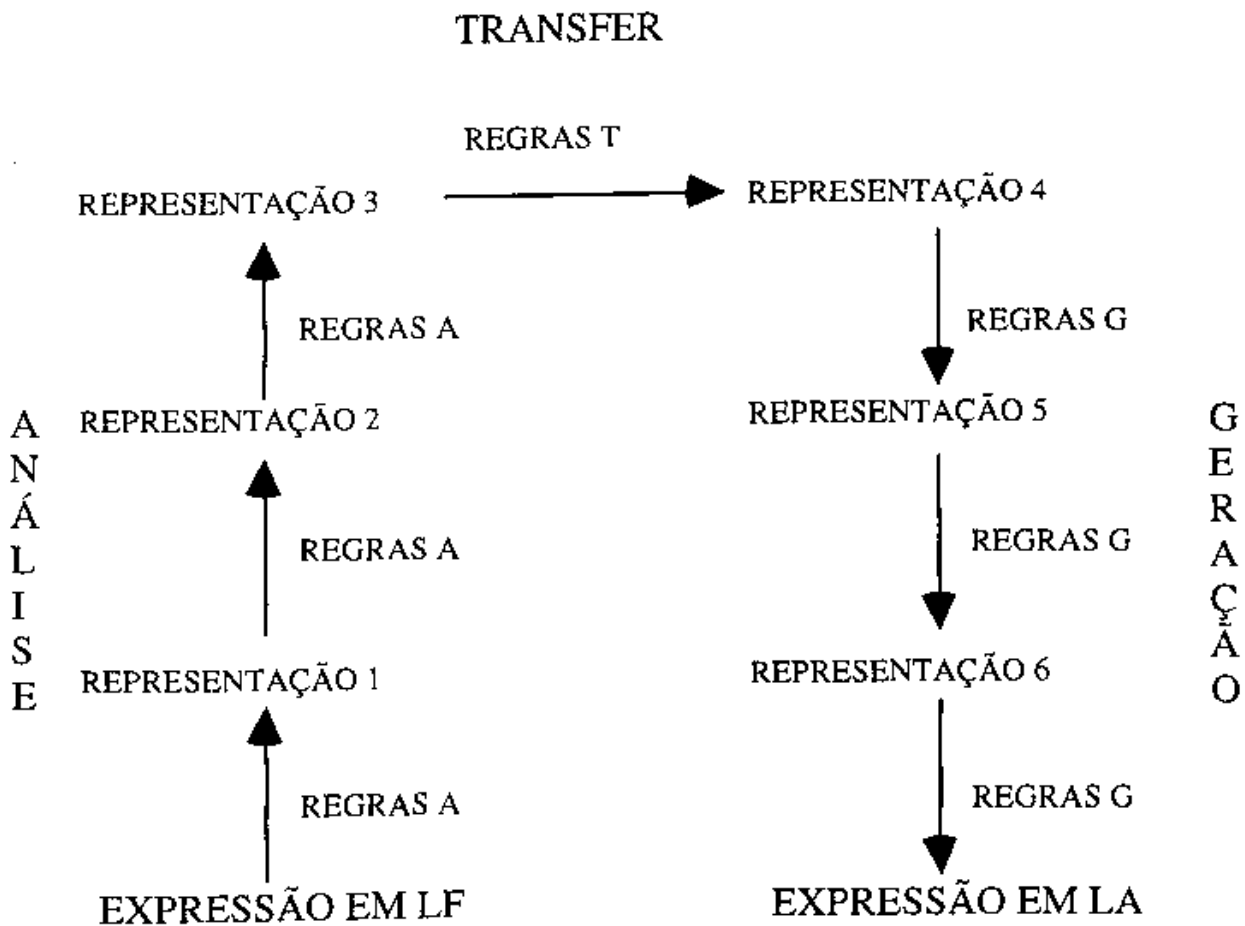
$$(2) T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4 \rightarrow T_5 \rightarrow T_6 \rightarrow T_7$$

O PROBLEMA DA ORDEM DOS MODIFICADORES ADJECTIVAIS

Supondo que T_1 e T_7 correspondem, respectivamente, a T_F e a T_A , uma das operações intermédias corresponderá necessariamente à operação que transfere a informação linguística contida numa expressão da língua de origem para a componente de geração do sistema (isto é, para o sistema de regras que utilizará tal informação para construir uma expressão linear bem-formada na língua de chegada). Uma regra de transfer projecta, portanto, uma expressão pertencente à gramática de partida (o lado esquerdo da regra) numa expressão pertencente à gramática de chegada (lado direito da regra).

A arquitectura de um sistema de TA utilizando o modelo de 'transfer' pode ser representada da seguinte forma:

(3)



A optimização de um sistema de transfer passa pela adopção de um 'princípio de conformidade' das representações linguísticas nos níveis de interface. Considerando as estruturas que alimentam as regras de transfer, estas devem ser representações canónicas, contendo, nomeadamente, informação de natureza semântica. Idealmente, tais represen-

tações não deveriam conter informação específica da língua de partida; no entanto, e daqui deriva o problema que vamos tratar, as limitações e propriedades formais das linguagens usadas na escrita das gramáticas fazem com que não seja possível filtrar toda a informação idiossincrática.

No que diz respeito às representações pertencentes à língua alvo (a expressão descrita no lado direito das regras de transfer e todas as expressões dos níveis subsequentes), convém ter presente que, dado que as expressões lineares (o nível terminal das representações produzidas) se caracterizam por propriedades — por exemplo, uma dada ordem de palavras — que são, por natureza, específicas da língua, é necessário reconstituir a informação relevante a partir da informação transferida. Tal reconstituição pode ser efectuada nas regras de transfer, através do sistema de projecção definido, como por exemplo, ao fazer o 'transfer' de itens lexicais, no caso de não existir correspondência entre a grelha temática do item em cada uma das línguas. Outra possibilidade consiste em atribuir às regras de síntese a tarefa de reconstituir a informação requerida; as regras de síntese podem inserir elementos de acordo com a informação contextual estipulada ou lexicalizar informação presente na estrutura sob a forma de traços sintácticos ou semânticos.

Estas duas estratégias para lidar com a informação linguística idiossincrática não são concorrenciais, tratando-se antes de alternativas usadas de acordo com o tipo de fenómeno a que se aplicam e com as limitações da implementação usada.

Passamos em seguida a apresentar algumas características particulares do módulo de 'transfer' usado no projecto GLEP.

As regras de 'transfer' implementadas na plataforma ALEP projectam uma Estrutura Linguística (LS) pertencente a uma gramática numa Estrutura Linguística Parcial (PLS) pertencente a outro.

O formato geral de uma regra de transfer, é o seguinte⁷:

(4) trule (<Especificador1>, <Especificador2>, <Direcção>, <Expressão1>, <Expressão2>, [<Condições>,]).

Supondo o caso da tradução de inglês para português, este formato corresponderá ao seguinte (representação parcial):

(5) trule (en:{ }, pt:{ }, =>, sign:{sem=> SEM1 }, sign:{sem=> SEM2 }, {SEM1=> SEM2}).

Certos elementos da regra dizem respeito a parâmetros da operação a aplicar: os valores do atributo 'specifier' ('en' e 'pt') que identificam a gramática de origem e a gramática alvo respectivamente; e o operador '=>' que define a direcção da projecção das regras (neste caso, a regra transfere apenas a expressão do lado esquerdo para a expressão no lado direito). Por outro, a regra contém informação de natureza linguística, especificando o tipo de estruturas a que se aplica e as condições de aplicação. Note-se que

as expressões de transfer <ExpressãoN> podem ser constituídas por diferentes entidades (LS, PLS, tipos, termos, variáveis, listas).

Esta regra emparelha ('match') com uma estrutura de input cujo nó de topo seja do tipo 'sign', extrai o valor do atributo 'sem' e transfere-o para produzir uma expressão linguística.

Uma vez apresentado o formato básico das regras de transfer, passaremos a referir as estratégias disponíveis para desenhar o módulo de 'transfer' num sistema baseado na plataforma ALEP⁸.

A estratégia mais utilizada no desenvolvimento dos projectos GSP e GLEP é a do chamado 'transfer simples', através de regras gerais que se aplicam a categorias em combinação com regras de transfer específicas para cada item lexical ou através de 'transfer encaixado', em que uma regra transfere a parte geral da informação e uma segunda regra, chamada pela primeira, transfere informação específica.

Um segundo tipo de 'transfer' é o chamado 'transfer por traços' ('featurized transfer') utilizado nos casos em que se pretende bloquear informação sintáctica presente na expressão de origem; por exemplo, no caso do 'transfer' de SNs importa transferir a informação semântica respeitante à determinação (genérico vs. específico, definido vs. indefinido, etc) mas não à realização sintáctica em inglês.

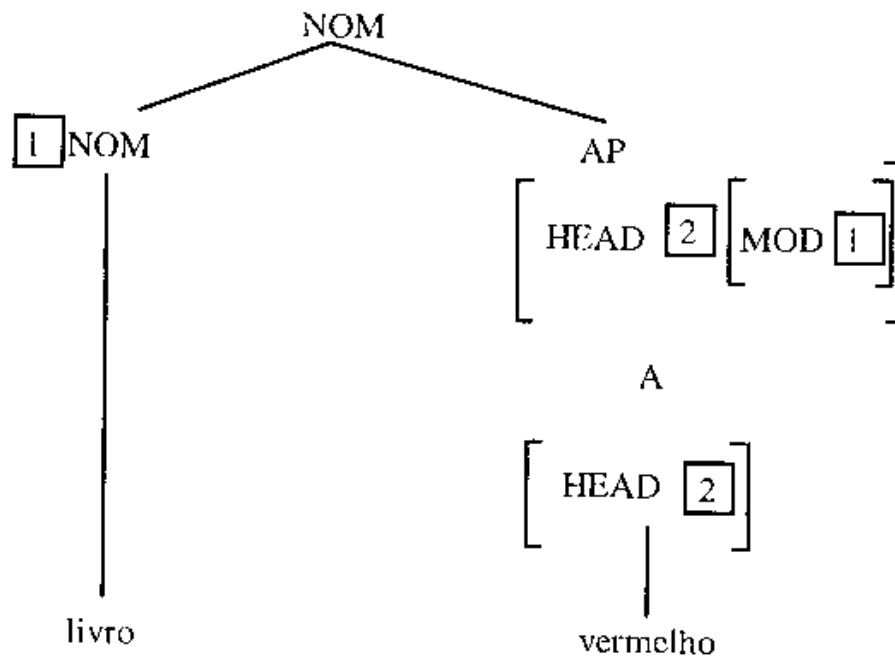
Certas categorias funcionais, como os complementadores e as preposições 'strongly bound', não são transferidas, sendo posteriormente inseridas por regras de síntese.

Finalmente, importa aqui considerar um caso particular de 'transfer encaixado', utilizado para lidar com modificadores; no caso destas estruturas, é necessário transferir informação representada sob a forma de listas, o que levanta o problema da obtenção da ordem correcta no português. A estratégia adoptada para resolver este problema constitui o objecto da segunda parte da presente comunicação⁹.

Antes, porém, importa apresentar de forma genérica o tratamento das estruturas de modificação proposto pela teoria da HPSG.

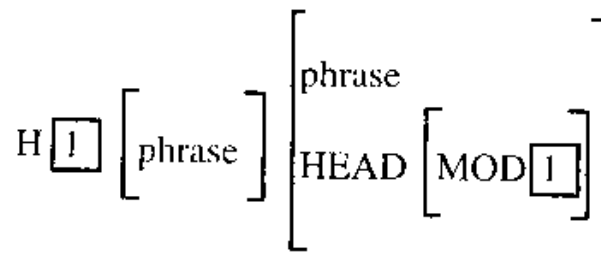
Segundo o modelo proposto por Pollard & Sag (1994: 55 sgs), os adjuntos seleccionam os núcleos que modificam. Como se sabe, neste modelo a informação é codificada sob a forma de estruturas de traços (isto é, pares atributo - valor) e a relação de modificação é expressa através de um traço de núcleo (HEAD feature): MOD (modified). A representação simplificada de um SN com um modificador adjectival será a seguinte (cf. Sag & Wasow (1997)):

(6)



De acordo com o HFP (HEAD FEATURE PRINCIPLE)¹⁰, o valor de MOD é percolado para o nó de topo da construção, tal como descrito na seguinte regra:

(7)



A ideia básica, no caso do tratamento de modificadores, quer em Pollard & Sag (1987) quer em Pollard & Sag (1994)¹¹, consiste em concatenar no nó de topo do NP, sob forma de lista, toda a informação sobre os elementos que o modificam. Mais concretamente, a definição de um objecto correspondente a um NP modificado conterá, dentro da sua especificação semântica (o que corresponde ao conjunto de especificações introduzidas sob o traço 'content'), uma lista de restrições (o traço 'restr' definido como de tipo lista) que concatenará um conjunto de objectos: esses objectos (eles próprios de tipo 'content') correspondem à especificação semântica de cada um dos modificadores do nome e à especificação do próprio nome.

Dado que o ALEP é construído como plataforma de implementação de gramáticas de tipo HPSG, o output da análise de estruturas em inglês reflecte precisamente as propostas de Pollard & Sag (1994). Nos objectos que podem ser output desse módulo de análise, a informação sobre modificação nominal é concatenada na lista 'restr' codificada como traço do nó de topo de um NP; numa frase, essa informação é codificada na lista 'restr' (lista de restrições) do argumento relevante do verbo principal. Deve notar-se que, na lista de restrições no output deste módulo de análise, qualquer sequência de adjectivos precederá o nome que modifica, o que será relevante na concepção das regras de síntese.

Ora o problema que aqui levantamos é precisamente o da geração da ordem de palavras, neste caso centrado no problema específico da geração da ordem de modificadores adjectivais e nome no interior de um NP. Podemos pensar que, tendo como ponto de partida uma lista de modificadores numa língua (o inglês) em que o adjectivo precede sempre o nome, a geração, em português, de um NP modificado por um adjectivo exige a definição de uma estratégia específica que permita gerar a ordem correcta.

Theofilidis, Verhagen & Badia (1994: 22-3), que traçam um conjunto de orientações para uma especificação semântica comum em gramáticas construídas em ALEP, notam desde logo o problema. A solução que sugerem para o problema parte do princípio de que a diferença na colocação dos adjectivos é captável em termos de pertença a diferentes classes semânticas e consiste numa complexificação do tipo 'restr'. Estes autores propõem que sejam declarados diferentes subtipos do tipo 'restr', correspondendo cada um destes subtipos a classes identificadas com valores semânticos (cf. (8)).

```
(8) type(restr:{{grad_restr=>list (type({content:{{{}}}), 'graduation'), (mann_restr=>
list (type({content:{{{}}}), 'manner'),(meas_restr=>list (type({content:{{{}}}),
'measure'), ...}}, 'restrictions').
```

[THEOFILIDIS, VERHAGEN & BADIA (1994: 23)]

A concretização desta proposta seria, porém, problemática: seria possível que precisássemos, no módulo de síntese, de tantas regras de formação de estruturas de modificação (em Pollard & Sag (1994), 'head-adjunct schema') quantos os subtipos declarados para o tipo 'restr'.

Além disso, o problema parece poder ser significativamente reduzido: Cinque (1996), embora colocando o problema num quadro teórico distinto (o da gramática generativa), argumenta a favor de uma ordem básica comum às línguas românicas e germânicas. No caso particular da ordem de constituintes no interior de um NP (DP, na sua análise), Cinque (1996) defende o movimento para a esquerda do nome nas línguas românicas e a existência de uma ordem fixa para os adjectivos - a variação na ordem de adjectivos e nomes nas línguas românicas por confronto com as línguas germânicas dirá portanto respeito à posição dos adjectivos em relação ao nome e não à posição relativa dos próprios adjectivos (em (9) ilustramos este último ponto).

(9) Ordem em inglês: A1 A2 A3 N

Possíveis ordens em português:

- i. N A1 A2 A3
- ii. A1 N A2 A3
- iii. A1 A2 N A3
- iv. A1 A2 A3 N
- v. A1 A3 N A2
- vi. A2 A1 N A3

Parece-nos portanto que, ao contrário do previsto por Theofilidis, Verhagen & Badia (1994), não será necessário declarar diferentes listas de restrições, na medida em que a ordem dos adjectivos na lista 'restr' será suficiente para predizer a ordem relativa dos adjectivos em português. O único problema a resolver será decidir até que ponto a lista dos adjectivos deve ser saturada (gerada, portanto) à esquerda - note-se que a saturação de listas se faz por saturação recursiva da cabeça e concatenação da cauda da lista.

Para resolver este problema, propomos dois novos subtipos do tipo 'inst_psoa' (o tipo usado na codificação do conteúdo semântico dos adjectivos): 'inst_psoa_left' e 'inst_psoa_right'. Em (10) apresentamos a hierarquia de tipos resultante desta alteração e que corresponde a uma alteração à hierarquia proposta em Theofilidis, Verhagen & Badia (1994). Pode parecer deselegante incluir, como subtipos do tipo semântico dos adjectivos, indicadores da sua distribuição; contudo, se reconhecermos que a distribuição dos adjuntos é condicionada pelo seu conteúdo semântico, os subtipos 'inst_psoa_left' e 'inst_psoa_right' podem (e devem) ser vistos como duas grandes classes correspondendo cada uma a um grupo de subtipos semânticos.

(10)

```

content > {
  neg_cont,
  pos_cont > {
    para_cont,
    psoa > {
      rel_psoa > {
        inst_psoa > {
          inst_psoa_left,
          inst_psoa_right },
        arg_psoa > {
          arg1_psoa,
          arg234_psoa > {
            arg2_psoa,
            arg34_psoa > {
              arg3_psoa,
              arg4_psoa } } } },
        name_psoa,
        tense_psoa,
        aspect_psoa },
      rd_cont > {
        r_index > {
          r_prop,
          r_npro,
          r_pron > {
            r_ppro,
            r_anap > {
              r_refl,
              r_recp } } } },
        r_psoa },
      lq_cont,
      cp_cont } }.

```

Voltemos agora às operações que, no ALEP, serão necessárias à geração de estruturas em português. Em primeiro lugar, é necessário assegurar a transferência correcta (operação de transfer) da informação relevante na língua de partida (o inglês) para uma estrutura que possa ser utilizada na geração da língua de chegada (o português). É precisamente na operação de transfer que é conseguida a identificação de cada adjectivo numa lista de restrições com um dos subtipos de 'inst_psoa'. Um adjectivo que, em português, ocorra à

esquerda será identificado com (transferido para) um objecto definido como 'inst_psoa_left'; um adjectivo que ocorra à direita será identificado com um objecto de tipo 'inst_psoa_right'; finalmente, um adjectivo que possa ocorrer tanto à esquerda como à direita, será identificado com um objecto do supertipo 'inst_psoa' - neste caso, a operação de síntese assegurará um output ambíguo.

No transfer de NP, mantivemos uma estratégia já testada de encaixe de regras. Assim, foi codificada uma única regra para transferir o conteúdo semântico de NP (cf. (11)). As regras específicas utilizadas para transferir cada adjectivo contido na lista de restrições são apresentadas em (12).

```
(11)
trule(en:{},pt:{},<=>,
      lq_cont:{
        quants=>QUANTS,
        rd_cont=>r_npro:{
          indx=>INDX=>ind_index:{
            agr=>AGR,
            gender=>_},
          restr=>RESTR}},
      lq_cont:{
        quants=>QUANTS,
        rd_cont=>r_npro:{
          indx=>INDX=>ind_index:{
            agr=>AGR, gender=>_},
          restr=>RESTR1}},
      [RESTR==RESTR1]).
```

(12)

```

trule(en:{ },pt:{ },<=>,
  lq_cont:{
    rd_cont=>r_adj:{
      indx=>INDX=>ind_index:{
        agr=>AGR, gender=>_},
      restr=>[inst_psoa:{
        rel=>rel:{
          rel_name=>big}}]}},
    lq_cont:{
      rd_cont=>r_adj:{
        indx=>INDX=>ind_index:{
          agr=>AGR, gender=>_},
        restr=>[inst_psoa:{
          rel=>rel:{
            rel_name=>grande}}]}},
    [] ).

```

```

trule(en:{ },pt:{ },<=>,
  lq_cont:{
    rd_cont=>r_adj:{
      indx=>INDX=>ind_index:{
        agr=>AGR,
        gender=>_},
      restr=>[inst_psoa:{
        rel=>rel:{
          rel_name=>african}}]}},
    lq_cont:{
      rd_cont=>r_adj:{
        indx=>INDX=>ind_index:{
          agr=>AGR,
          gender=>_},
        restr=>[inst_psoa_right:{
          rel=>rel:{
            rel_name=>african}}]}},
    [] ).

```

Transferida a informação da língua de partida, as regras de síntese assegurarão a geração em português. No caso específico dos modificadores, a partir do momento em que os adjectivos são identificados em português com os subtipos 'inst_psoa_left' e 'inst_psoa_right', a construção de regras de síntese não levanta problemas de maior. Temos portanto apenas duas regras para dar conta da modificação adjectival, uma que dá conta de adjectivos que ocorrem à esquerda do nome (cf. (13)), outra que dá conta de adjectivos que ocorrem à direita do nome (cf. (14)). Os subtipos do tipo 'inst_psoa' declarados para os adjectivos restringem a aplicação de cada uma destas regras: a regra em (13) só satura objectos na lista de restrições codificados como 'inst_psoa_left', a regra em (14) só pode saturar objectos codificados como 'inst_psoa_right'. Os objectos codificados com o supertipo 'inst_psoa' podem obviamente ser saturados por ambas as regras, obtendo-se assim um resultado ambíguo (resultado de facto pretendido).

A ordem de aplicação das regras não levanta qualquer problema, na medida em que é automaticamente restringida pela ordem dos adjectivos na lista de restrições. De acordo com as observações de Cinque (1996), se há adjectivos que devam ocorrer à esquerda, esses adjectivos serão também os que acabarão por ocorrer nos primeiros lugares da lista de restrições e, conseqüentemente, os que serão saturados em primeiro lugar. Todas as aplicações da regra 'adj_n' em (13) precederão as aplicações da regra 'n_adj' em (14), nos casos de estruturas complexas de modificação.

(13)

ld:{

spec => m_SPEC_PS[adj_n],

sign => lexical:{

str_key=>key:{

m_key=>KEY},

string => string:{

first => CONC_A,

rest => CONC_C},

synsem =>synsem:{

syn => syn:{

category => head_cat:{

head =>HEAD=>

n_head:{

modifies =>

modif_no:{ } } },

content =>lq_cont:{

quants=>[],

rd_cont=>r_npro:{

indx=>ind_index:{

```

    agr=>A,
    gender=>B },
    restr=>[lq_cont:{
      rd_cont=>r_adj:{
        indx=>ind_index:{ },
        restr=>[
          inst_psoa_left:{
            rel=>rel:{
              rel_name=>NAME}}]}]}|
      TAIL]]}}}}

```

<

[ld:{

```

    sign=> lexical:{
      string=>string:{
        first => CONC_A,
        rest=>CONC_B },
      synsem=> synsem:{
        syn =>syn:{
          category => head_cat:{
            head=> a_head:{ } }},
          content => lq_cont:{
            rd_cont=>r_adj:{
              indx=>ind_index:{
                agr => A,
                gender=>B },
              restr=>[inst_psoa_left:{
                rel=>rel:{
                  rel_name=>NAME}}]}]}]}},

```

ld:{

```

    sign => lexical:{
      str_key=>key:{
        h_key=>KEY=>n},
      string => string:{
        first => CONC_B,
        rest => CONC_C },
      synsem =>synsem:{

```

```

syn => syn:{
  category => head_cat:{
    head =>HEAD=>
    n_head:{
      modifies =>
      modif_no:{ } } },
content =>|q_cont:{
  quants => |],
  rd_cont=>r_npro:{
    indx=>ind_index:{
      agr=>A,
      gender=>B},
    restr=>
    TAIL}}}}}|.

```

(14)

```

ld:{
spec => m, SPEC_PS[n_adj],

sign => lexical:{
  str_key=>key:{
    m_key=>KEY},
string => string:{
  first => CONC_A,
  rest => CONC_C},
synsem =>synsem:{
  syn => syn:{
    category => head_cat:{
      head =>HEAD=> n_head:{
        modifies =>
        modif_no:{ } } } },
content =>|q_cont:{
  quants=>|],
  rd_cont=>r_npro:{
    indx=>ind_index:{
      agr=>A,
      gender=>B},
    restr=>|q_cont:{
      rd_cont=>r_adj:{
        indx=>ind_index:{ },
        restr=>|
        inst_psoa_right:{

```

```

rel=>rel:{
  rel_name=>NAME}}}}{TAIL}}}}

```

```

< [
ld:{

```

```

  sign => lexical:{
    str_key=>key:{
      h_key=>KEY=>n},
    string => string:{
      first => CONC_A,
      rest => CONC_B},
    synsem =>synsem:{
      syn => syn:{
        category => head_cat:{
          head =>HEAD=>
          n_head:{
            modifies =>
            modif_no:({}) }},
        content =>lq_cont:{
          quants => [],
          rd_cont=>r_npro:{
            indx=>ind_index:{
              agr=>A,
              gender=>B},
            restr=> TAIL}}}}},

```

```
ld:{

```

```

  sign=> lexical:{
    string=>string:{
      first => CONC_B,
      rest=>CONC_C},
    synsem=> synsem:{
      syn =>syn:{
        category => head_cat:{
          head=> a_head:({})},
        content => lq_cont:{
          rd_cont=>r_adj:{
            indx=>ind_index:{
              agr => A,

```

```

gender=>B },
restr=>[inst_psoa_left:{
  rel=>rel:{
    rel_name=>NAME}]]]]]]].

```

Deve ainda dizer-se que, na construção destas regras de síntese, não seguimos absolutamente as orientações contidas em Pollard & Sag (1994): nas regras de formação de estruturas de modificação nominal implementadas na gramática do GLEP, o nome (e não o adjectivo) é a cabeça da regra. Tal como dissemos anteriormente, na lista de restrições codificada no output do módulo de análise, qualquer sequência de adjectivos precederá o nome modificado. Consequentemente, se saturamos (na operação de síntese) essa lista de restrições da esquerda para a direita (tomando recursivamente a cabeça da lista como o elemento a ser saturado), a saturação será guiada pela informação codificada no nó nominal e acabará quando a definição desse nó nominal unificar com a entrada lexical do nome em causa (o que só pode acontecer quando todos os adjectivos tiverem sido saturados).

Mas existem consequências menos imediatamente previsíveis de uma tal resolução da síntese de estruturas de modificação - essas consequências prendem-se directamente com o que acontece no caso da saturação de adjectivos à direita do nome. Acontece que, se obtivermos, através da operação de transfer, uma lista de restrições contendo três adjectivos e um nome, numa sequência como a que é apresentada em (15) abaixo, não obteremos em síntese uma sequência como a apresentada em (16i) - aquela que seria esperada num sistema como o de Cinque (1996) -, mas obteremos uma sequência como em (16ii).

(15) A1 A2 A3 N

(16)

(16i) A1 N A2 A3

(16ii) A1 N A3 A2

Este tipo de situação parece dificilmente contornável, na medida em que resulta da interacção das possibilidades de saturação de listas e de características inerentes à operação de síntese. Concretamente,

- (i) a saturação de listas aplica-se da esquerda para a direita (i.e., consiste necessariamente na saturação recursiva da cabeça da lista);
- (ii) as estruturas de síntese são construídas de cima para baixo.

Assim, numa lista contendo uma sequência como em (15), o adjectivo A2, sendo saturado antes do adjectivo A3, acaba como o adjectivo mais à direita. De facto, se uma regra de construção de estrutura de modificação saturar o adjectivo A2 construindo uma

estrutura de nome + adjectivo e a operação de síntese continuar, expandindo o nó nominal noutra estrutura de nome + adjectivo, no output final A3 precederá A2, embora ambos correctamente sigam o nome.

E eis que, ao contrário do esperado, nem todas as consequências desta operação são indesejáveis. Existem, de facto, não poucos casos em que a ordem linear esperada é (9ii) e não (9i) - estes casos são aqueles que Cinque (1996: 304-5) aponta como aparentes contra-evidências à sua análise. Como exemplo desses casos, Cinque apresenta a versão francesa dos NP em inglês transcritos em (17i). Em (17ii) apresentamos as versões francesa e portuguesa dos mesmos NP:

(17i)

- a) a huge orange fruit
- b) a delicious cold chicken
- c) a handicapped elderly person

(17ii)

- a) un fruit orange énorme
um fruto laranja enorme
- b) un poulet froid délicieux
um frango frio delicioso
- c) une personne âgée handicapée
uma pessoa idosa deficiente

Cinque (1996) dá conta destes casos assumindo que o segundo dos adjectivos que ocorre à direita em francês e em português não é um adjectivo atributivo, mas antes um adjectivo predicativo. Acontece que uma tal distinção, embora possível no quadro teórico da HPSG (veja-se a referência ao traço [PRD -] em Pollard & Sag (1994: 55)), seria difícil de manter na gramática de geração que desenvolvemos em ALEP. Por um lado, uma distinção desse tipo é uma distinção fundamentalmente semântica que teria de ser captada na língua de partida e transferida para a língua de chegada - ora acontece que não só esta distinção não é captável pelo sistema tal como ele existe, como também as construções na língua de partida serão na maioria ambíguas. Por outro lado, a complexificação da gramática de síntese necessária à manutenção de tal distinção seria demasiada e as perdas em termos de tempo de processamento não seriam compensadas pela melhoria significativa do output, especialmente porque em muitos casos, em português, a ordem relativa dos adjectivos ocorrendo à direita pode ser alterada, não provocando por isso agramaticalidade (cf. (18))¹².

- (18) um cão gordo branco
um cão branco gordo

Mantivemos no GLEP a solução aqui descrita para as estruturas de modificação de NP envolvendo adjectivos, precisamente porque verificámos ser essa a solução que conduz aos melhores resultados (em termos de um maior número de resoluções acertadas) dentro da plataforma de implementação em que trabalhamos.

NOTAS:

1. O projecto GLEP (Gramática de Larga Escala do Português) é financiado pelo programa PRAXIS XXI (PCSH/C/CLCB 123/96).
2. Cf Theofilidis, Verhagen & Badra (1994).
3. O ALEP (Advanced Language Engineering Platform) é uma plataforma para o desenvolvimento de ferramentas linguísticas, nomeadamente gramáticas computacionais, desenvolvida pela Cray Systems sob encomenda da Comissão Europeia. Caracteriza-se por ter uma arquitectura aberta e é constituída por diversos módulos, implementados em diferentes linguagens, nomeadamente C e Prolog.
4. O projecto LSGram (Large Scale Grammars) teve como objectivo a construção de gramáticas de análise usando a plataforma ALEP, envolvendo várias línguas europeias. O projecto GSP (Gramática de Síntese do Português), financiado pelo programa Eureka / JNICT (EU 676 EUROLANG) tinha como objectivo investigar a possibilidade de desenvolvimento de um sistema de tradução com base na mencionada plataforma. Em resultado deste projecto foi construído um sistema experimental que constitui a base para a construção do protótipo a desenvolver no âmbito do projecto GLEP.
5. Por comodidade de exposição, é referido o caso simples de um par de línguas: o que aqui é dito poderá ser facilmente alargado ao caso mais complexo de *n* línguas. Igualmente, o que se refere a propósito das operações de análise poderia ser alargado com as devidas adaptações ao caso do processo simétrico de geração.
6. Nos modelos reais, devido às propriedades da implementação, as regras de transfer podem ser formalmente distintas de outros tipos de regras; como veremos adiante, é este o caso do sistema GLEP.
7. Cf Simpkins (1994:15-17)
8. Cf. *Documentação do projecto GSP*
9. Um outro tipo de transfer, 'transfer estrutural' ou 'complexo' poderia igualmente ser mencionado. No presente estado de desenvolvimento, o GLEP não faz uso de regras deste tipo, as quais são necessárias para lidar com casos em que o tipo de informação a transferir depende de propriedades da estrutura.
10. "HEAD FEATURE PRINCIPLE: In a headed phrase, the values of SYNSEM | LOCAL | CATEGORY | HEAD and DAUGHTERS | HEAD-DAUGHTER | SYNSEM | | LOCAL | CATEGORY | HEAD are token-identical". Pollard & Sag (1994: 399).
11. Embora a ideia geral aqui descrita de tratamento de modificadores seja comum a Pollard & Sag (1987) e Pollard & Sag (1994), existe uma diferença crucial entre as propostas contidas nestas duas referências: enquanto Pollard & Sag (1987) assumem que os núcleos seleccionam os adjuntos que os modificam, Pollard & Sag (1994) defendem já que os adjuntos, sendo cabeças semânticas nas estruturas de adjunção, seleccionam os núcleos que modificam. Para uma síntese das duas propostas, veja-se Pollard & Sag (1994: 55-57).
12. Note-se que este tipo de observação não parece ser válido para todas as línguas românicas, pelo que uma estratégia de geração como a que propomos não poderá ser tão facilmente generalizada. Em francês, a ordem dos adjectivos à direita parece ser mais fixa, como mostra o conjunto de frases apresentado abaixo (os juízos são apenas os de um falante):
 J' ai vu un chien gras blanc.
 ?? J' ai vu un chien blanc gras.

BIBLIOGRAFIA

- G. CINQUE (1996). On the evidence for partial N-movement in the Romance DP. In *Italian Syntax and Universal Grammar*, New York: Cambridge University Press.
- Documentação do projecto GSP, ILTEC, 1997.
- CARL POLLARD & IVAN A. SAG (1987). *Information-Based Syntax and Semantics*, Volume 1 - Fundamentals. CSLI.
- CARL POLLARD & IVAN SAG (1994). *Head-Driven Phrase Structure Grammar*, Chicago & London: CSLI/The University of Chicago Press.
- IVAN SAG & THOMAS WASOW (1997) *Syntactic Theory: A Formal Introduction*, Draft, Center for the Study of Language and Information, Stanford.
- SIMPKINS (1994). ET-6/1 Linguistic Formalism - ALEP 2 User Guide. Luxembourg: CEU.
- AXEL THEOFILIDIS, MARC VERHAGEN & TONI BADIA (1994). Specifications for a Common Semantic Representation Format, LS-GRAM Deliverable D-WP1a