DA OBSERVAÇÃO DE PROPRIEDADES LINGUÍSTICAS À SUA FORMALIZAÇÃO NUMA GRAMÁTICA DO PROCESSAMENTO DA LÍNGUA[1]

Caroline Hagège (caroh@gril.univ-bpclermont.fr, cah@iltec.pt)
Gabriel G. Bès (gabriel@gril.univ-bpclermont.fr)

GRIL (Groupe de Recherche dans les Industries de la Langue).
Université Blaise Pascal
34, av. Carnot
63000 Clermont-Ferrand
FRANCE

Resumo

Neste artigo pretende-se apresentar uma nova metodologia para a expressão das propriedades linguísticas do português, com vista à utilização destas propriedades numa gramática para processamento automático da língua. Uma das vantagens desta abordagem, ao contrário do que acontece com as gramáticas de unificação, é a sua total independência dos algoritmos utilizados no processamento informático.

Outra das vantagens é a facilidade de manipulação dos axiomas de descrição, o que vai simplificar o trabalho de rectificação e de afinação das propriedades. Para o processo de rectificação e afinação, foram desenvolvidas duas ferramentas informáticas: o verificador de axiomas e o gerador de modelos. Estas ferramentas permitem testar, a partir de *corpora*, a validade dos axiomas, ao confrontá-los com sequências linguísticas encontradas nos textos. Finalmente, as propriedades linguísticas acima descritas vão poder ser exploradas de maneira sistemática, com vista à sua integração numa gramática, seja esta uma gramática de unificação, ou uma gramática de superfície.

Neste artigo são apresentados os axiomas de descrição assim como as ferramentas de verificação acima mencionadas. Será exemplificada a utilização e validação dos axiomas num sub-conjunto de categorias do sintagma nominal nuclear em português europeu. Em conclusão, veremos como é possível recuperar a partir dos axiomas as propriedades linguísticas, formalizando-as numa gramática do processamento automático do português.

Motivações

Qualquer trabalho linguístico rigoroso, seja ele de linguística teórica ou de linguística computacional, deve ser baseado na observação dos fenómenos da língua. Aplicando à linguística uma metodologia científica tradicional, um trabalho que tenha como objectivo o processamento automático de uma língua deve ter em conta os seguintes fundamentos:

1) a observação dos dados
2) a formulação de hipóteses sobre o modo como os dados são organizados
3) a verificação dessas hipóteses

Acreditamos que ultrapassados todos estes passos, uma vez consolidadas as bases, se pode passar a uma etapa seguinte:

4) A reutilização das propriedades na escrita de uma gramática *HPSG* para o processamento de uma língua.

O artigo aqui apresentado está centrado nos pontos 2) e 3), sendo também abordada a passagem para a quarta etapa acima descrita.

Os pressupostos subjacentes ao trabalho aqui apresentado são os seguintes:

1) Existem regularidades e restrições na utilização da linguagem natural (LN)

[1] Trabalho desenvolvido no quadro do doutoramento em linguística computacional da primeira autora, financiado pela FCT(Fundação para a Ciência e Tecnologia, Praxis XXI), realizado no GRIL, sendo o ILTEC a instituição de acolhimento.
Agradecimentos à Ermelinda Gonçalinha, Luísa Marques da Silva Coheur, Fernando Leite e Carla Diogo pelas suas releituras e comentários.

Por outras palavras, não se pode falar ou escrever agrupando de um modo aleátorio elementos do vocabulário.

Por exemplo, no português europeu, se for utilizada a palavra *ambos* para quantificar um nome, esta deverá ser seguida por um artigo definido.

ambos os rapazes

**ambos rapazes*

**ambos uns rapazes*

Repare-se ainda que na sequência *ambos os rapazes*, não se pode utilizar um cardinal para quantificar *rapazes.*

* *ambos os três rapazes*

Existem muitas propriedades simples que podem ser expressas em termos de:

- exigências entre certas categorias (no primeiro exemplo de *ambos*, exigência de um artigo definido)
- exclusões entre certas categorias (no segundo exemplo de *ambos* não é permitido o cardinal)

2) Dificuldade de acesso à informação linguística nas gramáticas de unificação

Nas gramáticas de unificação utilizadas frequentemente hoje em dia no processamento da LN (como por exemplo, *UCG*, *HPSG* e *LFG*), a informação puramente linguística não é facilmente acessível por várias razões:

- A informação linguística encontra-se "diluída" em vários locais

Por exemplo, a ordem linear em *HPSG* é parcialmente indicada nas regras de concatenação de constituintes (os esquemas de dominância imediata), mas também o é dentro do léxico, sob a forma de listas ordenadas correspondentes à sub-categorização de uma dada entrada.

- Existe uma fronteira pouco clara entre a informação puramente linguística e a informação necessária à boa execução do programa.

Nas gramáticas de unificação que pretendem ter alguma cobertura, para além da informação linguística, há sempre um conjunto de traços sem realidade linguística (que são no fundo pequenos "truques") que têm por fim "pôr a gramática a funcionar". Este tipo de informação, inevitável na construção de uma gramática minimamente poderosa, não pertence ao domínio do conhecimento linguístico, mas sim ao do algoritmo e dificulta grandemente a leitura, a manutenção e a reutilizabilidade destas gramáticas, pois estes artefactos encontram-se ao nível da descrição linguística.

Estas duas razões têm como consequência dificultar a extensão e recuperação da informação numa gramática *HPSG* para extender esta gramática ou recuperar somente o conhecimento linguístico aí contido ou ainda para rescrever outra gramática noutro tipo de formalismo.

Qualquer linguista que pretenda extrair as propriedades da língua a partir duma gramática da qual não é o autor, mesmo conhecendo o formalismo em questão, vai ter dificuldades nesta tarefa.

Isto constitui sem dúvida um problema para a utilização dos formalismos de unificação.

3) Distinção entre estrutura dos objectos linguísticos e relações entre estes objectos

Acreditamos na necessidade de distinguir os objectos linguísticos das relações que existem entre estes objectos. Com efeito, nas teorias linguísticas actuais, tal distinção não existe (propagação dos traços numa gramática HPSG por exemplo). Na nossa abordagem, não existem pressupostos quanto à forma dos objectos manipulados numa língua. O que nos interessa aqui são as relações que podem existir entre os objectos independentemente das suas estruturas. Serão elaboradas uma série de hipóteses sobre estas relações expressas sobre a forma de propriedades, propriedades essas que serão relacionadas com o contexto sintáctico e semântico onde aparecem estes objectos.

Proposta para a representação do conhecimento linguístico independentemente dum formalismo gramatical particular

Foi desenvolvida no GRIL uma metodologia para poder expressar o conhecimento linguístico de uma forma simples, expressiva, e independente de qualquer formalismo gramatical.[1]

Se considerarmos como unidades significativas (US) aquelas que recebem uma etiqueta morfossintáctica (podem tratar-se de compostos, por isso não se fala de *palavra*), verificamos que as ligações entre estas US não são sempre do mesmo tipo.

Por exemplo, no caso de *o meu gato bebeu leite* as ligações entre as palavras *o* e *meu*, e *meu* e *gato* são de natureza diferente do que a ligação entre *gato* e *bebeu*.

[1] Trabalho iniciado por G. G. Bès e continuado por G. G. Bès e C. Hagège.

Estes trechos dentro dos quais as US têm uma relação privilegiada, vão ser designados por sintagmas nucleares (SXN[1]), estendendo-se desde o princípio até ao núcleo deste sintagma.

Por exemplo, na frase citatada anteriormente, a sequência de US *o meu gato* constitui um SNN (sintagma nominal nuclear), pois começa no princípio[2] do sintagma e acaba no núcleo *gato.*

Dentro dos SXN, existem certas propriedades formais importantes:

- não existe recursividade dentro dos sintagmas nucleares, exceptuando o fenómeno da coordenação;
- a linguagem que os descreve, salvo algumas excepções, tem a particularidade de ser aceite por um autómato K1F (*i.e*, um autómato que só necessita de conhecer o estado anterior para se dirigir para o estado seguinte);
- existem relações de dependência que poderão ser uma base para o cálculo da semântica entre as US de um SXN.

Tendo em vista estas particularidades foi elaborada uma série de axiomas que permitem descrever os sintagmas nucleares duma língua de um modo exaustivo e pormenorizado.

Existem três tipos de axiomas que vão ser apresentados e exemplificados na descrição de um (pequeno) sub-conjunto do SNN sujeito do português europeu. Estes axiomas são:

- os axiomas de existência;
- os axiomas de linearidade;
- os axiomas de dependência.

Convenções utilizadas

Um símbolo x vai subsumir um símbolo x_s e x_p, sendo estes últimos símbolos respectivamente utilizados pela forma singular e pela forma plural de x.

O símbolo @ a seguir a uma categoria significa que esta deve ser considerada como núcleo.

O símbolo º a seguir a uma categoria significa que esta não é considerada como núcleo.

Um símbolo x subsume os símbolos x@ e xº.

[1] S para sintagma, N para nuclear e X para um dos possíveis valores do núcleo destes sintagmas.

[2] Entende-se como princípio, ou o início de uma frase, ou a primeira US que ocorre após o final de outro sintagma nuclear.

1- Axiomas de existência

1-1 Subsunção entre elementos do vocabulário : predicado **remplace/2**

remplace(<categoria>,[<lista de categorias>]).

Descrição do axioma
A categoria <categoria> subsume as categorias listadas em <lista de categorias>.
Exemplo de aplicação
1) remplace(n,[nc,npr]). 2) remplace(nc,[nc_s,nc_p,nc1,nc2]). 3) remplace(nc_s,[nc1_s,nc2_s]). 4) remplace(nc1_s,[nc1_m_s,nc1_nm_s]). 5) remplace(nc_p,[nc1_p,nc2_p]). 6) remplace(nc1_p,[nc1_m_p,nc1_nm_p]). 7) remplace(nc1,[nc1_s,nc1_p]). 8) remplace(nc2,[nc2_s,nc2_p]). 9) remplace(npr,[npr1,npr2,npr3]). 8) remplace(adj,[adj1,adj2,adj3,adj_s,adj_p]). 9) remplace(adj1,[adj1_s,adj1_p]). 10) remplace(adj2,[adj2_s,adj2_p]). 11) remplace(adj3,[adj3_s,adj3_p]). 12) remplace(adj_s,[adj1_s,adj2_s,adj3_s]). 13) remplace(adj_p,[adj1_p,adj2_p,adj3_p]).
Comentário em português
• O símbolo n (para nome) é uma generalização dos símbolos nc (nome comum) e npr (nome próprio). • Além das formas singular e plural dos nomes comuns, são consideradas duas grandes classes de nomes comuns, nc1 e nc2 (os nomes contáveis e os nomes massivos). • A classe dos nc1 é dividida em duas sub-classes (os nomes contáveis quantificáveis por *todo* e os não quantificáveis por *todo.* • Os nomes próprios são organizados em três subclasses que correspondem à possibilidade ou não de serem precedidos por um determinante. • Os adjectivos são dividos em três subclasses que correspondem aos adjectivos pré- e pós-nominais que podem ser núcleos, aos adjectivos pré- e pós-nominais que não podem ser núcleos e aos adjectivos exclusivamente pós-nominais.

1-2 Declaração do vocabulário : predicado amod/2

amod(<id> , [<lista de categorias ou modelos>]).

Descrição do axioma
Qualquer sequência <id> é subsumida por um elemento da lista. Qualquer elemento da lista subsume pelo menos um elemento de uma sequência gramatical <id>.
Exemplo de aplicação
amod(snn-pt,[n, adj, card_p, poss, dem, arti_s, pri, todo, tal, uns_p, ambos_p]).
Comentário em português
Dentro das sequências possíveis do snn português, haverá uma ou mais categorias (ou sub-categorias) da lista amod.

1-3 Expressão do núcleo : predicado obligdi/2

obligdi(<id> , [<lista de categorias ou modelos>]).

Descrição do axioma
Uma sequência <id> tem um e só um elemento subsumido por um elemento da lista. Qualquer categoria ou modelo da lista subsume pelo menos um elemento duma sequência.
Exemplo de aplicação
obligdi(snn-pt, [n, pri, adj, card_p, poss, arti_s, dem, todo_p, ambos_p, uns_p, tal]).
Comentário em português
As categorias da lista poderão ser núcleo do snn português. Nota-se aqui que as únicas categorias previamente declaradas que não podem ser núcleo são todo_s e artd. Com efeito, consideramos nesta descrição que nem a forma *todo* (ou *toda*) nem as formas *o*, *a*, *os*, e *as* podem ser núcleo de um snn sujeito.

1-4 Unicidade dos elementos do vocabulário : predicado **uniq/2**

uniq(<id> , [<lista de categorias ou modelos>]).

Descrição do axioma
Uma sequência <id> que tem um elemento pertencendo a <lista de categorias ou modelos> só tem este elemento uma vez.
Exemplo de aplicação
uniq(snn-pt,[n, adj, card_p, poss, artd, dem, pri, todo, ambos_p, tal, uns_p]).
Comentário em português
Qualquer sequência snn do português não poderá ter duas ou mais vezes uma das categorias da lista. Nota-se que a coordinação não entra na nossa descrição.

1-5 Exigências introduzidas por certas categorias : predicado **exig/2**

exig(<id>, [<lista 1> , <lista de listas de cat ou modelos>]).

Descrição do axioma
Uma sequência <id> que contém <lista1> também deve ter incluída pelo menos uma das listas da lista de listas do segundo argumento.
Exemplo de aplicação
1) exig(snn-pt, [[nc_s], [[artd], [arti_s], [dem], [tal]]]). 2) exig(snn-pt,[[npr2],[[artd],[dem]]]). 3) exig(snn-pt,[[npr1],[[neant],[todo_s]]]). 4) exig(snn-pt,[[npr1,todo_s],[[neant]]]). 5) exig(snn-pt,[[pri],[[neant]]]). 6) exig(snn-pt,[[poss],[[artd],[dem]]]). 7) exig(snn-pt,[[adj_s]], [[artd], [arti_s],[dem], [tal]]]). 8) exig(snnpt,[[adj,poss],[[nc]]]). 9) exig(snn-pt,[[todo_p],[[neant],[artd], [dem]]]). 10) exig(snn-pt,[[todo_s],[[artd],[dem], [npr1]]]). 11) exig(snn-pt,[[ambos_p],[[neant],[artd]]]).
Comentário em português
Se o ou os elementos da primeira sub-lista estão presentes num snn do português, este snn deverá também necessariamente ter pelo menos um ou mais elementos do vocabulário indicados na segunda sub-lista. • O primeiro axioma de exigência estipula que um nome comum no singular deve necessariamente ser acompanhado por um dos determinantes artigo definido, indefinido, demonstrativo ou tal. Isto obviamente não vai impedir a presença de outros elementos do vocabulário, i.e a sequência *todo o rapaz* não contradiz o axioma. • O segundo axioma dá conta da obrigatoriedade de um determinante com os nomes próprios de tipo 2. • O terceio axioma indica que um nome próprio de tipo 1 (ex. *Portugal*) ou está sozinho num snn do português, ou então pode estar presente com todo (ex. *todo Portugal*). • O axioma seguinte indica que na nossa descrição, se *todo* e um nome próprio de

tipo 1 estão presentes num snn do português, então este snn não poderá conter mais nenhuma categoria.

- No quinto axioma vemos que a categoria designada por pri , não pode estar acompanhada por nenhuma outra categoria dentro de um snn. Este axioma vai dar conta de snn do tipo *alguém.*
- O sexto axiomo formaliza o facto de que o emprego de um possessivo em português europeu implica o emprego de um artigo definido ou de um demonstrativo.
- Um adjectivo no singular utilizado dentro de um snn exige um dos determinantes da segunda sub-lista.
- O oitava axioma permite dar conta da impossibilidade seguinte :
  * a minha amigável
  embora seja possível aceitar
  a minha amigável companheira
- Os dois axiomas seguintes indiquem a obrigatoriedade de artigos definidos ou demonstrativos com as formas flexionadas de todo. Nota-se que na forma singular também é possível ter um nome próprio de tipo 1.
- O último axioma exprime as exigências de *ambos*/*ambas* que pode constituir um snn sem a presença de outras categorias ou que exige a presença de um artigo definido

Definimos o caso particular de uma lista contendo o símbolo "néant" (nada) que vai permitir exprimir que um sub-conjunto de símbolos (eventualmente reduzido a um só símbolo) constitui por ele próprio um snn.

1-6 Exclusões introduzidas por certas categorias : predicado **exclu/2**

exclu(<id> ,[<lista de listas de cat ou símbolos>]).

Descrição do axioma
Uma sequência <id> não pode ter duas listas especificadas como elementos do segundo argumento.
Exemplo de aplicação
1) exclu(snn-pt,[[ambos_p], [todo], [tal]]). 2) exclu(snn-pt,[[dem],[artd],[arti_s], [uns_p]]). 3) exclu(snn-pt,[[uns_p],[poss],[arti_s]]). 4) exclu(snn-pt,[[card_p],[ambos_p]]). 5) exclu(snn-pt,[[card_p],[ambos_p],[nc2]]). 6) exclu(snn-pt,[[card_p],[tal@]]). 7) exclu(snn-pt, [[uns_p],[ambos_p]]).
Comentário em português
Se o ou os elementos do vocabulário presentes na primeira sub-lista aparecem dentro de um snn do português , então não poderão aparecer dentro deste snn nenhuns dos elementos do vocabulário contidos nas outras sub-listas • O primeiro axioma indica que um snn quantificado por *ambos* não pode ser quantificado por *todos*, nem pode conter a palavra *tais*. • O segundo axioma exprime o facto de que só pode ocorrer um único determinante para determinar o núcleo do snn. • O terceiro axioma indica que não podem ocorrer dentro de um mesmo snn o possessivo e o artigo indefinido. • O axioma seguinte exprime a exclusão entre um cardinal (ex. *três*) e *ambos*. • O quinto axioma dá conta do facto de que nem *ambos* , nem um cardinal podem quantificar um nome massivo. • O sexto axioma exprime a restricção seguinte : * *as três tais* sem no entanto impedir sequências correctas como : *as três tais mulheres* *as tais três mulheres* *as tais três* Nota-se aqui a utilização do símbolo @ • O último axioma impede que *uns* e *ambos* ocorrem juntos dentro de um mesmo snn.

2- Axiomas de linearidade

Relações de ordem do vocabulário dentro de um modelo : predicado **precede/3**

precede(<id>, <S>, <lista de categorias ou modelos>).

Descrição do axioma
Num modelo <id>, S deve preceder qualquer elemento da lista do terceiro argumento, sendo S um símbolo que represente uma categoria ou um modelo.
Exemplo de aplicação
1) precede(snn-pt,todo,[quelconque]). 2) precede(snn-pt,ambos_p,[quelconque]). 3) precede(snn-pt,artd,[poss,card_p,tal,n,adj]). 4) precede(snn-pt,dem,[poss,card_p,tal,n,adj]). 5) precede(snn-pt,arti_s,[n,adj]).

6) precede(snn-pt,uns_p,[card_p,adj,n]). 7) precede(snn-pt,card_p,[adj,n]). 8) precede(snn-pt,tal,[adj,n]). 9) precede(snn-pt,adj,[n]). 10) precede(snn-pt,x°,[x@]).
Comentário em português
O segundo argumento de precede/3 deve estar à esquerda de qualquer categoria presente no terceiro argumento de precede/3. É utilizado o símbolo "quelconque" que subsume qualquer categoria. Nota-se no décimo axioma a indicação do facto de que na nossa descrição, uma categoria não núcleo precede sempre uma categoria núcleo, i.e o núcleo do snn é na nossa descrição a última categoria à direita.

3- Axiomas de dependência

Relações de dependência para o cálculo da relações semânticas : predicado fleche/3

fleche(<id>,<S1>,<S2>).

Descrição do axioma
Indica que existe uma relação de dependência entre S1 e S2, sendo S1 e S2 símbolos de categoria ou modelo.
Exemplo de aplicação
fleche(snn-pt,xº,x@).
Comentário em português
Este predicado exprime a propriedade seguinte : dentro de um snn do português, todas as categorias não-núcleo estabelecem uma relação de dependência com o núcleo.

As propriedades de dependência não vão só existir dentro do sintagma nuclear mas também vão permitir estabelecer dependências entre os vários sintagmas nucleares construídos.

Importante : o conjunto dos axiomas não constitui uma gramática do português, mas sim uma forma de explicitar regularidades linguísticas entre os símbolos utilizados.

4- *Exemplo de aplicação dos axiomas*

Os seguintes exemplos mostrem como a utilização dos axiomas podem validar ou invalidar certas sequências snn do português.

- a sequência *ambos* não entra em contradição com nenhum dos axiomas de exclusão, e esta sequência obedece a uma das restrições do axioma de exigência 11.
- a sequência *ambos os* não constitui um snn pois entra em contradição com o axioma de precedência 10 (a categoria artigo definido não pode ser núcleo, ver o axioma obligdi).
- a sequência *ambos os rapazes* não entra em contradição com nenhuns dos axiomas e obedece a uma das retricções do axioma de exigência 11.
- A sequência *os* não pode constituir um snn do português pois viola o axioma obligdi.
- A sequência *ambos uns rapazes* entra em contradição com o axioma de exigência 11.

Ferramentas de avaliação das propriedades

São duas as ferramentas informáticas desenvolvidas para avaliação da pertinência das propriedades face aos dados linguísticos reais :

- O verificador de axiomas
- O gerador de modelos

O verificador de axiomas é um programa desenvolvido em *Prolog* que, com base em axiomas de existência e de linearidade para a descrição dos sintagmas nucleares e em sequências extraídas manualmente de *corpora* e etiquetadas com o etiquetador morfo-sintáctico SMORPH[1], cujas etiquetas correspondem às categorias utilizadas pelos axiomas, vai permitir verificar em que medida os axiomas estão, ou não, em conformidade com a realidade linguística proveniente dos textos.

Caso haja contradição entre os axiomas e o texto, a localização da contradição é feita indicando qual é o axioma que não está em conformidade com uma dada sequência. Deste modo, é possível diagnosticar a propriedade errada e corrigi-la.

[1] SMORPH é um programa genérico de tokenização e etiquetagem morfo-sintáctica concebido e desenvolvido no GRIL por Salah Aït-Mokhtar. A aplicação ao Português foi desenvolvida por C.Hagège (ver [Ait97], [Ait98] e [Hagège97]).

O gerador de modelos (em desenvolvimento por G. Bès cf. [Bès98]) , como o seu nome indica, permite gerar, a partir dos axiomas, todos os modelos definidos pelos axiomas e apenas estes. Este gerador de modelos permite aceder aos modelos deduzidos a partir dos axiomas a três níveis diferentes : um nível que apresente conjuntos (não ordenados) das categorias que constituem os modelos, outro nível que corresponde à aplicação nestes conjuntos das regras de precedência (obtenção de listas ordenadas de categorias) e enfim, um nível no qual aparecem, além da ordem linear, as relações de dependência. A obtenção dos modelos vai permitir validar ou invalidar os axiomas, i.e. validar ou invalidar as hipóteses feitas sobre a língua.

Importante: O verificador de axiomas e o gerador de modelos não são "parsers" para o processamento do português, mas sim ferramentas de avaliação e de auxílio ao linguísta para a formalização das propriedades de uma língua.

Os axiomas, uma vez devidamente validados com os instrumentos de que dispomos, exprimem de uma forma concisa e clara (não é necessário conhecer nenhum formalismo particular, nem é necessário ser um linguista experimentado para compreender e explorar estes dados) as propriedades linguísticas de uma linguagem natural.

Neste momento foi feita a descrição pormenorizada do sintagma nominal nuclear do português europeu. Foi feito pelo segundo autor uma descrição do domínio verbal do francês. Nestes dois casos foi possível exprimir todas as restricções necessárias às descrições utilizando a axiomática acima apresentada.

Exploração dos axiomas com vista a sua utilização numa gramática do processamento do português.

Após a apresentação da metodologia de formalização do conhecimento linguístico, explicita-se como é possível recuperar este conhecimento.

Foram efectuadas duas experiências a partir da descrição do SNN português: uma com uma gramática de superfície, outra com uma gramática de unificação *HPSG* (versão [Sag97]). Falaremos da segunda experiência, mostrando como, a partir de algumas regras simples, se pode inferir uma série de constatações que vão ajudar à constituição do léxico da gramática (sendo o léxico a componente central numa gramática *HPSG*).

1- Utilização do gerador de modelos

O gerador de modelos apresentado acima, além de oferecer uma ferramenta de validação e controlo das propriedades linguísticas, vai permitir extrair contextos de emprego de uma dada categoria .

Ao escrever-se uma gramática *HPSG* deve contar-se com certos pré-requisitos linguísticos, herdados da tradição gramatical. Com efeito, espera-se que uma unidade como "*belas*" tenha uma etiqueta de adjectivo e não de artigo numa sequência como *belas raparigas*. Além de mais, o próprio formalismo impõe uma série de pressupostos que devem ser tomados em conta.

Convenções utilizadas:

- *lista_us* vai designar uma sequência de US que constituem um snn e que é possível derivar graças ao gerador de modelos.
- *head* vai designar o caminho SYNSEM:SYN:LOC:CAT:HEAD
- *spr* vai designar o caminho SYNSEM:SYN:LOC:CAT:VAL:SPR
- *spec* vai designar o caminho SYNSEM:SYN:LOC:CAT:HEAD:SPEC
- *el* vai designar uma entrada lexical HPSG

2- O que a teoria impõe

2-1 hierarquia de tipos

Qualquer signo linguístico possui no caminho SYNSEM:SYN:LOC :CAT o atributo HEAD de tipo head que tem os subtipos seguintes:

head
- subst
  - noun
  - verb
  - adj
- func
  - det
  - mark

2-2 Formato das entradas lexicais

- head:noun

spr:L

e L=[ ] ou L = [Y1,...Yn] e spr:det subsume Y1 e Yn, signos linguísticos

- head:det

spr:[ ]

spec:Z
e head:noun subsume Z, sendo Z um signo linguístico

- head:adj

spr:[ ]
spec: [ ]
modif : W e W signo linguístico subsumido por head:noun

3- O que se pode deduzir da observação das listas_us

Regra 1
Todas as formas que podem surgir o mais à direita numa lista_us deverão ter uma entrada lexical com head:noun.[1]

Regra 2
Todas as formas utilizadas isoladamente numa lista_us deverão ter uma entrada lexical com spr:[ ]

Regra 3
Todas as formas que começam à esquerda uma lista_us, que não são consideradas como adjectivos e que não são utilizadas só numa lista_us, deverão ter uma el cuja descrição deve ser subsumida por head.det, spec:W, sendo W um signo linguístico subsumido pour head:noun. Estas formas na lista_us serão notadas f_det (forma det).

Regra 4
Se numa lista_us existem f_deti e f_detj nesta ordem uma das possibilidades para a construção do léxico é:
1) para a forma mais à direita na lista (para a qual já se definiu através da regra 1 uma el com head:noun), é necessário acrescentar na el correspondente o traço spr:[det2,det1], sendo deti um signo linguístico subsumido por head:det e sabendo que det2 corresponde ao signo linguístico que descreve f_deti e det1 corresponde ao signo linguístico que descreve f_detj.
2) vão ser criados dois subtipos do tipo inicial det, det_a e det_b
3) vão ser especificados det1 e det2 que vão ter respectivamente os valores det_a e det_b para o traço head.

[1] O facto de decidir se esta entrada nominal é directamente codificada no léxico ou obtida através de uma regra lexical é deixado ao critério do linguista.

Regra 5
Qualquer forma que se encontre entre o especificador e o núcleo é considerado como adjectivo.
Neste caso o el correspondente vai conter os traços head:adj e modif:Z sendo Z subsumido por head:noun.

Conclusão
Nesta comunicação foi feita a proposta de uma metodologia para a explicitação do conhecimento linguístico. Esta metodologia baseia-se na definição de um sistema axiomático que permite exprimir as propriedades de uma língua sem presupor da forma dos objectos linguísticos manipulados. O importância dada à verificação das hipóteses linguísticas é uma das características desta metodologia que tenta considerar a linguística como uma ciência empírica. Foi também esboçado o modo como se pode utilizar o conhecimento linguístico assim formalizado numa gramática de unificação *HPSG*. A recuperação deste conhecimento linguístico numa gramática deste tipo torna-se mais difícil pois temos de lidar com pressupostos linguísticos da própria teoria. Foram feitas outras experiências para a utilização dos dados linguísticos numa gramática de superfície, sendo neste caso possível explorar de uma forma sistemática os axiomas para a construção da gramática.

Referências

[Ait97] Aït-Mokhtar S. (1997) "Du texte ASCII au texte lemmatisé: la présyntaxe en une seule étape" in *Actes de TALN'97*. Grenoble.
[Ait98] Aït-Mokhtar S. (1998) *L'analyse présyntaxique en une seule étape*. Thèse de Doctorat en Linguistique et Informatique. Université Blaise Pascal – Clermont-Ferrand.
[Bès98] Bès G. G. (1998) *Le générateur de modèles*. Documento interno ao GRIL.
[Cunha85] Cunha C., Cintra L. (1985) *Nova Gramática do Português Contemporâneo*. Rio de Janeiro, Ed. Nova Fronteira.
[Eliseu97] Eliseu, A. Santos, A. Gonçalinho E. (1997) "O problema da ordem dos modificadores adjectivais em português e em inglês no contexto de um sistema de Processamento de Linguagem Natural bilingue" in *Actas do XIII da APL*, Lisboa (a publicar)

[Gärtner98] Gärtner E. (1998) *Grammatik der Portuguiesischen Sprache*, Tübingen, Max Niemeyer Verlag.
[Mateus89] Mateus M.H., Brito A.M. , Duarte I., Faria I. H. (1989) *Gramática da Língua Portuguesa*. 2a. edição revista e aumentada, Lisboa, Ed. Caminho.
[Pollard94] Pollard, C., Sag I. (1994) *Head-Driven Phrase Structure Grammar*.CSLI, Chicago: Chicago University Press.
[Sag97] Sag, I.A., Wasow T. (1997) *Syntactic Theory: A Formal Introduction*. CSLI, Partial Draft of September.