Universidade de São Paulo - USP
Universidade Federal de São Carlos - UFSCar
Universidade Estadual Paulista - UNESP

ConPor: um gerador de estruturas conceituais UNL

Lucia Specia
Lucia Helena Machado Rino

NILC-TR-02-15

Novembro, 2002

Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional
NILC - ICMC-USP, Caixa Postal 668, 13560-970 São Carlos, SP, Brasil

Resumo

Este relatório descreve o processo de desenvolvimento da primeira versão do **ConPor** (**Con**ceitualização do **Por**tuguês), um sistema cuja função é mapear estruturas sintáticas geradas pelo *parser* do NILC em estruturas conceituais UNL. São discutidas as questões que nortearam esse mapeamento, bem como sua implementação.

Este trabalho conta com o apoio
financeiro da CAPES

Índice

Figuras

Tabelas

1 Introdução

Sistemas de Processamento de Línguas Naturais (PLN) que realizam a interpretação da língua natural (LN) têm como objetivo extrair o significado de expressões da LN e exprimi-lo de acordo com algum modelo de representação do conhecimento independente de LN. Dependendo do nível de profundidade dessa interpretação, tais expressões podem corresponder a sentenças isoladas ou a discursos completos e a representação do significado pode envolver, portanto, apenas o conhecimento lingüístico (morfológico, sintático e semântico)[1] ou o conhecimento extralingüístico (conhecimento de senso comum, pragmático-discursivo, etc.).

Nesse sentido, existem diferentes níveis de representação do significado, sendo a representação semântica o nível mais básico e, assim, apenas uma parte dessa representação. Quanto mais profundo o nível de conhecimento envolvido, maior a complexidade para a obtenção automática de uma representação do significado. Por isso, grande parte dos sistemas de interpretação se limita ao processamento semântico, considerando apenas sentenças de sentido literal e isoladas do seu contexto de ocorrência. Esse é o cenário do sistema que será descrito neste relatório: abordaremos somente questões relativas à **representação semântica**, também chamada aqui de **representação conceitual**.

O modo como essa representação é obtida pode variar conforme a abordagem de interpretação utilizada pelo sistema. Normalmente, essa interpretação é dividida em etapas, cada qual responsável pelo processamento de um tipo de conhecimento, como o morfológico, o sintático e, por fim, o semântico. Em algumas abordagens, cada etapa de processamento é realizada independentemente das demais, assumindo como entrada os resultados da etapa anterior. Nesse caso, o processamento semântico realiza um mapeamento entre as estruturas criadas na fase de análise sintática e a representação do seu significado. Outras abordagens consideram uma interação entre as diferentes etapas, intercalando os processamentos sintático e semântico. Análises simultâneas ou intercaladas permitem que informações semânticas sejam utilizadas para verificar problemas na estrutura sintática, durante a sua criação, eliminando interpretações sintáticas que seriam semanticamente incorretas. Por outro lado, a vantagem de manter os estágios de processamento sintático e semântico separados é que o processamento sintático pode ser mais sofisticado, uma vez que é independente de domínio, diferentemente do processamento semântico. Além disso, haja vista que muitos analisadores sintáticos apresentam resultados bastante satisfatórios e estão disponíveis para uso, é possível concentrar esforços para resolver o problema da análise semântica.

As diferentes etapas de processamento podem utilizar modelos específicos para a representação do(s) conhecimento (s) que utilizam. Alguns modelos lingüístico-computacionais permitem a representação e o processamento de conhecimentos em diferentes níveis, como é o caso da Gramática Léxico-Funcional (*Lexical-Functional Grammar* – LFG) (Kaplan & Bresnan, 1982). No caso específico da representação do conhecimento semântico, existem diferentes teorias ou linguagens de representação semântica, como a Gramática de Casos (Fillmore, 1968), a Teoria de Dependência Conceitual (Schank, 1975), a Teoria da Semântica Conceitual (Jackendoff, 1990), a linguagem LCS (*Lexical Conceptual Structure*) (Dorr, 1992; Dorr, 1993) e a linguagem UNL (*Universal Networking Language*) (UNL, 2001).

[1] Estamos tratando aqui somente de sistemas para o processamento da língua escrita, portanto, não consideramos níveis de conhecimento para o processamento da língua falada, como o fonético-fonológico.

Nessas teorias ou linguagens de representação semântica, assim como nas teorias de significado de um modo geral, o significado de um texto é independente da forma como ele é representado, ou seja, da asserção lingüística desse texto, no entanto, esta pode ajudar a indicar e refletir o significado. Por exemplo, o relacionamento gramatical geralmente expressa, em maior ou menor grau, o relacionamento semântico entre os constituintes sentenciais. Por essa razão, algumas teorias ou linguagens de representação semântica, como a Semântica Conceitual e a LCS, prevêem e fornecem mecanismos para o mapeamento entre forma (estruturas sintáticas) e significado (estruturas semânticas ou conceituais). Outras, como a Gramática de Casos, prescrevem essa relação na direção contraria, ou seja, o mapeamento entre significado e forma.

Neste trabalho pretendemos descrever o sistema **ConPor** (**Con**ceitualização do **Por**tuguês), desenvolvido para realizar o mapeamento de estruturas sintáticas de sentenças da língua portuguesa em estruturas conceituais, com base na linguagem UNL como modelo de representação semântica. A arquitetura global do sistema, incluindo a descrição dos seus recursos lingüísticos e módulos de processamento e exemplos dos seus dados de entrada e saída, é apresentada na Seção 2. O *corpus* dos exemplos que delimitam as estruturas sintáticas previstas no mapeamento é apresentado na Seção 3. O processo de mapeamento, propriamente dito, é descrito na Seção 4. As questões relacionadas à implementação desse processo são apresentadas na Seção 5. Algumas conclusões e considerações finais sobre o trabalho são discutidas na Seção 6.

2 A arquitetura do sistema ConPor

A arquitetura do sistema ConPor é ilustrada na Figura 1. Dentre os recursos lingüísticos e processos utilizados por esse sistema, alguns já estão disponíveis no NILC[2] e são reutilizados aqui (módulos em azul), enquanto outros são desenvolvidos exclusivamente para o ConPor (módulos em vermelho). A seguir, serão descritos os componentes dessa arquitetura, bem como um exemplo do funcionamento do sistema.

2.1 Fase de Preparação

A **Fase de Preparação** representa um conjunto de componentes auxiliares responsáveis por fornecer a entrada efetiva para o processo de mapeamento entre estruturas sintáticas e conceituais. O principal componente desse módulo é o *parser* do NILC, uma ferramenta independente deste trabalho, desenvolvida para fins diversos, e utilizada aqui para minimizar o esforço do processo como um todo. Assim, na arquitetura da Figura 1, o módulo do *parser* do NILC representa uma abstração de um sistema bastante complexo, que também faz uso de recursos e subprocessos, como um léxico e um analisador lexical e morfológico, que não estão ilustrados aqui.

Nessa fase, o *parser* do NILC gera todas as estruturas sintáticas possíveis para uma sentença do português. Como não utiliza informações semânticas para o seu processamento, nem sempre todas as estruturas estão corretas e, na maioria dos casos, pode-se identificar apenas uma delas como mais apropriada. Portanto, ainda na Fase de Preparação, a estrutura sintática mais provável é escolhida manualmente.

A disponibilidade do *parser* do NILC como uma ferramenta bastante robusta e eficaz nos permitiu focalizar o processo de análise semântica. A falta de informações semânticas no

[2] Núcleo Interinstitucional de Lingüística Computacional (www.nilc.icmc.sc.usp.br).

processamento sintático foi resolvida, aqui, pela seleção manual de uma das estruturas sintáticas geradas.

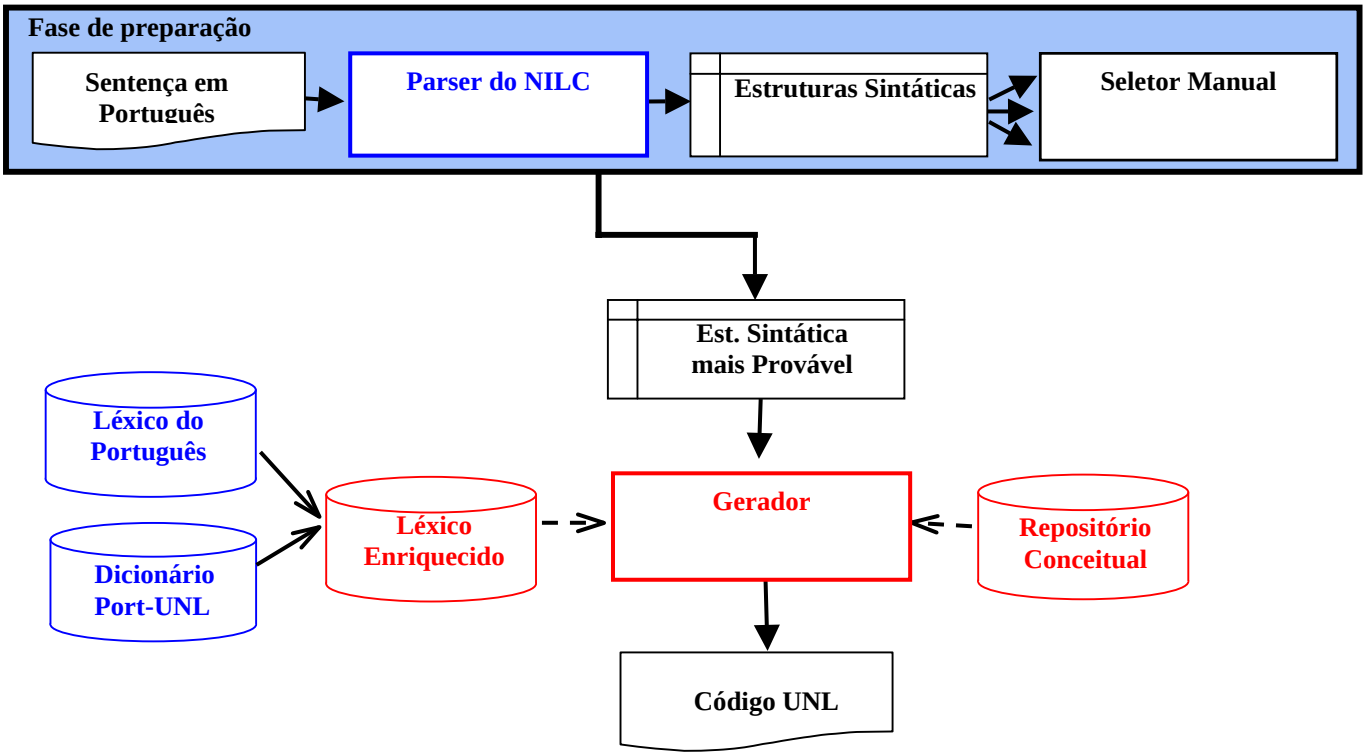


Figura 1 – Arquitetura do ConPor

2.2 Recursos

Os recursos reutilizados pelo sistema, que já haviam sido criados no NILC para outras aplicações, são o Léxico do Português (Nunes et al., 1996) e o Dicionário Português UNL (Dias-da-Silva et al., 1998). Neste trabalho, esses dois recursos fornecem informações para o desenvolvimento de um novo recurso, chamado **Léxico Enriquecido**.

O Léxico Enriquecido é um léxico semântico que armazena as informações morfossintáticas e semânticas das palavras das sentenças do *corpus* necessárias para o mapeamento das estruturas sintas sintáticas em estruturas conceituais. Portanto, além das informações recuperadas dos dois recursos citados, dispõe de informações específicas para essa tarefa. Uma descrição desse recurso pode ser consultada em Specia & Rino (2002)**.** O **Repositório Conceitual** armazena os *templates*, ou seja, os modelos para o mapeamento entre estruturas sintáticas e conceituais, conforme será descrito na Seção 4.

2.3 Gerador

O processo **Gerador** é responsável, efetivamente, pelo mapeamento das estruturas sintáticas em estruturas conceituais (**mapeamento sintático-conceitual**, doravante). Para tanto, esse processo assume como entrada a estrutura sintática escolhida como mais provável dentre as geradas pelo *parser* do NILC, controla a utilização dos *templates* do Repositório Conceitual, e também de alguns subprocessos para a edição das entradas (pré-edição) e das

saídas (pós-edição). Como resultado, é gerada uma estrutura conceitual chamada de **Código UNL** ou **Documento UNL**. Os detalhes desse módulo são apresentados nas Seções 4 e 5.

2.4 Linguagem de representação semântica

A linguagem utilizada para representar as estruturas conceituais do ConPor é a UNL (*Universal Networking Language*) (UNL, 2001). Essa linguagem foi desenvolvida com o propósito de ser implementada em sistemas de comunicação multilingual, mais especificamente, para ser utilizada como representação intermediária no Projeto UNL (Uchida et al., 1999), de tradução automática por interlíngua. Uma descrição dessa linguagem pode ser consultada em UNL (2001)[3].

A estrutura conceitual da UNL é uma representação lógica, baseada em relações semânticas binárias entre conceitos independentes de língua natural. Para tanto, a UNL se utiliza de três componentes: rótulos de relações semânticas (RLs – *Relation Labels*), palavras que indicam conceitos universais (UWs – *Universal Words*)[4] e rótulos de atributos que especificam ou restringem esses conceitos (ALs – *Attribute Labels*). Com base nesses componentes, uma estrutura conceitual UNL tem o seguinte formato:

$$RL(UW_1.@AL, UW_2.@AL)$$

Sendo que UW_1 e UW_2 devem ser diferentes e os ALs são opcionais e em número variável, sempre precedidos por '@'. Para representar todos os conceitos de uma sentença e as relações entre eles, portanto, é utilizado um conjunto de relações binárias dessa categoria, conforme mostra o exemplo a seguir.

2.5 Um exemplo

Como exemplo do processamento do sistema, incluindo a Fase de Preparação, considere a sentença (1) e sua correspondente estrutura conceitual UNL (2), gerada a partir da estrutura sintática produzida pelo *parser*, ilustrada na Figura 2[5].

(1) O Sol em Sagitário ilumina seus relacionamentos.

(2) agt(illuminate.@entry, sun.@def)
plc(sun.@def, sagittarius)
obj(illuminate.@entry, relationship.@pl)
pos(relationship.@pl, you)

Para representar conceitualmente a sentença (1) são utilizadas quatro relações semânticas (agt = agente; plc = lugar; obj = objeto; pos = posse) entre os conceitos *sun* (sol), *sagittarius* (sagitário), *illuminate* (ilumina) e *relationship* (relacionamentos), sendo que alguns desses conceitos têm atributos associados a eles, como *@pl* em *relationship*, que indica que o conceito é coletivo.

[3] Disponível em http://www.unl.ias.unu.edu.
[4] Apesar de ser independente de LN, a UNL utiliza palavras da língua inglesa como símbolos para expressar seus conceitos.
[5] Sendo: SUJ = sujeito, SN = sintagma nominal, AADNE = adjunto adnominal à esquerda, SDET = sintagma determinante, CN = complemento nominal, SP = sintagma preposicional, PREDV = predicado verbal, SVTD = sintagma verbal transitivo direto e OD = objeto direto.

```
(FRASE #O Sol em Sagitário ilumina seus relacionamentos#
   (PERIODO #O Sol em Sagitário ilumina seus relacionamentos#
      (PERIODO_INDEPENDENTE #O Sol em Sagitário ilumina seus relacionamentos#
         (SUJ #O Sol em Sagitário#
            (SUJ_SIMPLES #O Sol em Sagitário#
               (SN #O Sol em Sagitário#
                  (AADNE #O#
                     (SDET #O#
                        (nucleo #o# artigo)
                     )
                  )
                  (nucleo #sol# subst)
                  (CN #em Sagitário#
                     (SP #em Sagitário#
                        (nucleo #em# preposicao)
                        (SN #Sagitário#
                           (nucleo #sagitário# subst)
                        )
                     )
                  )
               )
            )
         )
         (PREDICADO #ilumina seus relacionamentos#
            (PREDV #ilumina seus relacionamentos#
               (SVTD #ilumina#
                  (nucleo #ilumina# verbo)
               )
               (OD #seus relacionamentos#
                  (OD_SIMPLES #seus relacionamentos#
                     (SN #seus relacionamentos#
                        (AADNE #seus#
                           (nucleo #seus# adj)
                        )
                        (nucleo #relacionamentos# subst)
                     )
                  )
               )
            )
         )
      )
   )
)
```

Figura 2 – Estrutura sintática da sentença (1)

Ao partir de estruturas sintáticas e não de sentenças em LN, o sistema é fortemente dependente dessas estruturas. Por essa razão, apresentaremos, na seção seguinte, algumas informações sobre o *corpus* de exemplos do ConPor, com base no qual foram definidas as estruturas sintáticas que podem ser mapeadas pelo sistema.

3 O *corpus* base

O ConPor contempla somente um domínio e um subgrupo de construções gramaticais do português como entrada, que constituem o conjunto de exemplos com base nos quais os recursos lingüísticos e processos foram definidos, ou seja, o ***corpus* de exemplos**, ou o ***corpus* base**. O domínio do *corpus* é o de textos de horóscopo, escolhido por apresentar construções

gramaticais e vocabulário simples, adequados uma primeira proposta de prototipagem do sistema.

Para compor o *corpus* base, foram coletados textos completos do horóscopo diário de um jornal *on-line*, durante 30 dias. A opção por uma única fonte pode ser justificada pela homogeneidade verificada no referido domínio, pois os textos de diferentes fontes seguem, em geral, vocabulário e estilo de escrita muito próximos.

Os textos coletados para o *corpus* base totalizam 360, cada um referente a um signo, em um determinado dia, com três sentenças, em média, cada qual com cerca de 15 palavras. Desses textos, foram selecionadas determinadas sentenças, com base em alguns critérios visando a uma redução quantitativa (em termos do número de sentenças) e qualitativa (em termos de complexidade lingüística dessas sentenças para a modelagem computacional). Em alguns casos, essa "seleção" envolveu alguma forma de alteração, para correção ou simplificação das sentenças. Os procedimentos aplicados para a seleção das sentenças foram os seguintes:

1) Os textos foram corrigidos, manualmente e com o auxílio do revisor ortográfico e gramatical (ReGra) (Nunes e Oliveira, 2000) do *Microsoft Word*.

2) Como critério de redução quantitativa, determinamos, num primeiro momento, que seriam selecionadas sentenças dos primeiros 36 textos, correspondentes aos horóscopos dos três primeiros dias dos textos coletados.

3) Para facilitar a eliminação das construções mais complexas, todos os 36 textos foram classificados de acordo com o índice de legibilidade *Flesch Reading Ease* do *Microsoft Word* adaptado para o português (Martins et al., 1996), que classifica um texto em uma escala de 0 a 100 pontos, conforme a Tabela 1.

Tabela 1 – Faixas de classificação do índice *Flesch Reading Ease*

Escore	**Classificação**
75 a 100	Muito fácil
50 a 75	Fácil
25 a 50	Difícil
0 a 25	Muito difícil

4) Foram selecionados os sete textos considerados "muito-fáceis", de acordo com os escores do índice *Flesch Reading Ease*.

5) Mesmo dentre os textos considerados "muito-fáceis", havia construções gramaticais complexas. Portanto, quando necessário, as sentenças foram alteradas (manualmente) para que se tornassem mais simples. O principal objetivo era reduzi-las a sentenças nucleares (orações simples), ou seja, compostas de um só verbo, ou, no máximo, locuções verbais. Para tanto, algumas construções (sinais de pontuação, como ponto-e-vírgula, dois-pontos; conjunções integrantes; locuções conjuncionais; e marcadores discursivos, como "isto é", "ou seja", etc.) passaram a funcionar como delimitadores de sentença, de forma a eliminar sentenças coordenadas e subordinadas. Alguns exemplos da simplificação realizada são ilustrados na Tabela 2. As sentenças complexas que não puderam ser simplificadas foram excluídas do *corpus* base.

Tabela 2 – Exemplos de sentenças simplificadas

Sentença original	**Sentença(s) simplificada(s)**
Tire do armário aqueles planos malucos que você não ousa levar adiante.	Tire do armário aqueles planos malucos.
Amanhã ocorre a Lua crescente, e a luta entre	Amanhã ocorre a lua crescente.

presente e passado se tornará ainda mais aguerrida.	A luta entre presente e passado se tornará mais aguerrida.
Você é quem sabe o tamanho de sua perna e até onde agüenta sentir dor.	Você sabe o tamanho de sua perna.

6) Mesmo após a simplificação das sentenças, algumas delas foram alteradas ou excluídas porque não foram reconhecidas corretamente pelo *parser* do NILC[6]. Aqui, as alterações consistiram, basicamente, da troca de algumas palavras não aceitas (estrangeirismos, por exemplo), ou cuja categoria gramatical era incorretamente reconhecida, e da inversão na ordem de algumas construções para a ordem considerada gramaticalmente correta pelo *parser*. A Tabela 3 ilustra exemplos de sentenças alteradas em função do *parser*, e exemplos de sentenças excluídas por não serem reconhecidas pelo *parser*.

Tabela 3 – Exemplos de sentenças alteradas ou excluídas em função de restrições do *parser*

Sentença original	**Sentença simplificada**
As horas delicadas passarão, assim, em relativo controle e paz.	As horas delicadas passarão em relativo controle e em relativa paz.
Amanhã ocorre a lua crescente.	A lua crescente ocorre amanhã.
Sentenças excluídas	
Os planetas estão apontando a hora de mudança pessoal.	
A luta entre presente e passado se tornará mais aguerrida.	

7) Algumas sentenças foram alteradas para resolver dependências contextuais (anáforas e elipses, por exemplo), as quais não são contempladas pela linguagem UNL, cuja representação conceitual é feita sentença a sentença. Essa alteração só foi realizada, no entanto, para os casos nos quais a sentença, se analisada isoladamente, fora do contexto, perderia o sentido. Nos demais casos, as referências foram mantidas.

Como resultado dessa seleção (alterações e exclusões), obtivemos 36 sentenças simples, com as quais conseguimos simplificar expressivamente o projeto dos recursos e processos da primeira versão do sistema ConPor. Como vimos, apesar do processamento do sistema ser baseado nesse *corpus* de sentenças, sua entrada efetiva consiste da estrutura sintática dessas sentenças. Nessas 36 sentenças, foram identificadas 33 estruturas sintáticas diferentes, as quais respondem, portanto, pela principal diversificação do *corpus* base. Essas 33 estruturas são ilustradas no Apêndice A.

Certamente, esse número de estruturas é pequeno, se considerarmos o uso efetivo do sistema na interpretação de sentenças, mesmo para o domínio escolhido. Contudo, o objetivo deste trabalho é propor um modelo de interpretação semântica, e não um sistema robusto. De qualquer forma, a implementação desse modelo, conforme será descrito na Seção 5, é organizada de forma a permitir a inclusão de novos *templates* e regras para o mapeamento para outras construções gramaticais, bem como a inclusão das diferentes palavras das sentenças que representam essas construções. Com isso, o *corpus* base deverá ser estendido nas próximas versões do sistema, com relação tanto ao número quanto à complexidade das estruturas sintáticas.

Na próxima seção descrevemos o modelo de mapeamento especificado para as diferentes estruturas sintáticas do *corpus* base.

[6] Esse problema já era, em parte, previsto, uma vez que o *parser* ainda está em processo de desenvolvimento. A versão utilizada data do início de 2001, desde então, várias alterações e melhorias foram implementadas no *parser*. Fixamos essa versão para o trabalho para evitar a necessidade de adaptação do sistema em função de alterações desse parser.

4 O mapeamento sintático-conceitual

O processo de mapeamento realizado no módulo Gerador do sistema ConPor consiste em combinar os dados da estrutura sintática de entrada com um dos diversos ***templates***, ou *frames* (Minsky, 1975), de mapeamento do sistema, por meio de um processo de unificação dividido em três etapas. Esses *templates* constituem o **Repositório Conceitual** do sistema e foram definidos com base nas estruturas do *corpus* e seus respectivos códigos UNL gerados manualmente. A estrutura dos *templates* e as diferentes etapas do mapeamento são descritas a seguir.

4.1 A estrutura dos *templates*

Os *templates* do ConPor representam e manipulam, por meio dos seus atributos, informações de diferentes naturezas. De acordo com o tipo de informação, os atributos podem ser divididos em:

1) Atributos morfossintáticos: incluem a estrutura sintática das sentenças e algumas informações morfossintáticas dos seus constituintes;
2) Atributos semânticos: incluem informações semânticas dos constituintes da sentença;
3) Atributos de correspondência: disparam processos que fazem a correspondência entre a função sintática dos constituintes da estrutura sintática e sua função semântica (papel semântico). Um constituinte, aqui, pode ser composto por mais de uma palavra, por exemplo, um sintagma nominal, composto por um artigo e um substantivo, é mapeado em um único papel semântico;
4) Atributos relacionais: disparam processos que estabelecem as relações semânticas da UNL entre os constituintes semânticos.

Desse conjunto de atributos, alguns possuem valores fixos, que devem combinar com as informações provindas da estrutura sintática sendo mapeada ou dos traços do Léxico Enriquecido para as suas palavras, e servem, portanto, como restrições para o mapeamento. Outros atributos possuem valores em aberto, que devem ser preenchidos com as informações da estrutura sintática ou do Léxico Enriquecido, e servem, portanto, para fornecer informações para o resultado ou para as fases posteriores do processo de mapeamento, como para a inserção de alguns rótulos de atributos (ALs) UNL, no caso dos atributos morfossintáticos.

Os diversos atributos são utilizados em vários momentos no processo de mapeamento sintático-conceitual. Seguindo uma arquitetura modular, a estrutura dos *templates* foi dividida em três partes, cada qual com determinados atributos. Como conseqüência, o processo de mapeamento foi dividido em três etapas, cada qual responsável por uma tarefa específica, conforme ilustra a Figura 3. Nessa figura, são representados os atributos que estabelecem algumas restrições (em vermelho) e os atributos que fornecem, quando a unificação é bem sucedida, as informações (em verde), para a etapa seguinte (no caso das duas primeiras etapas), ou o resultado final do mapeamento (no caso da última etapa).

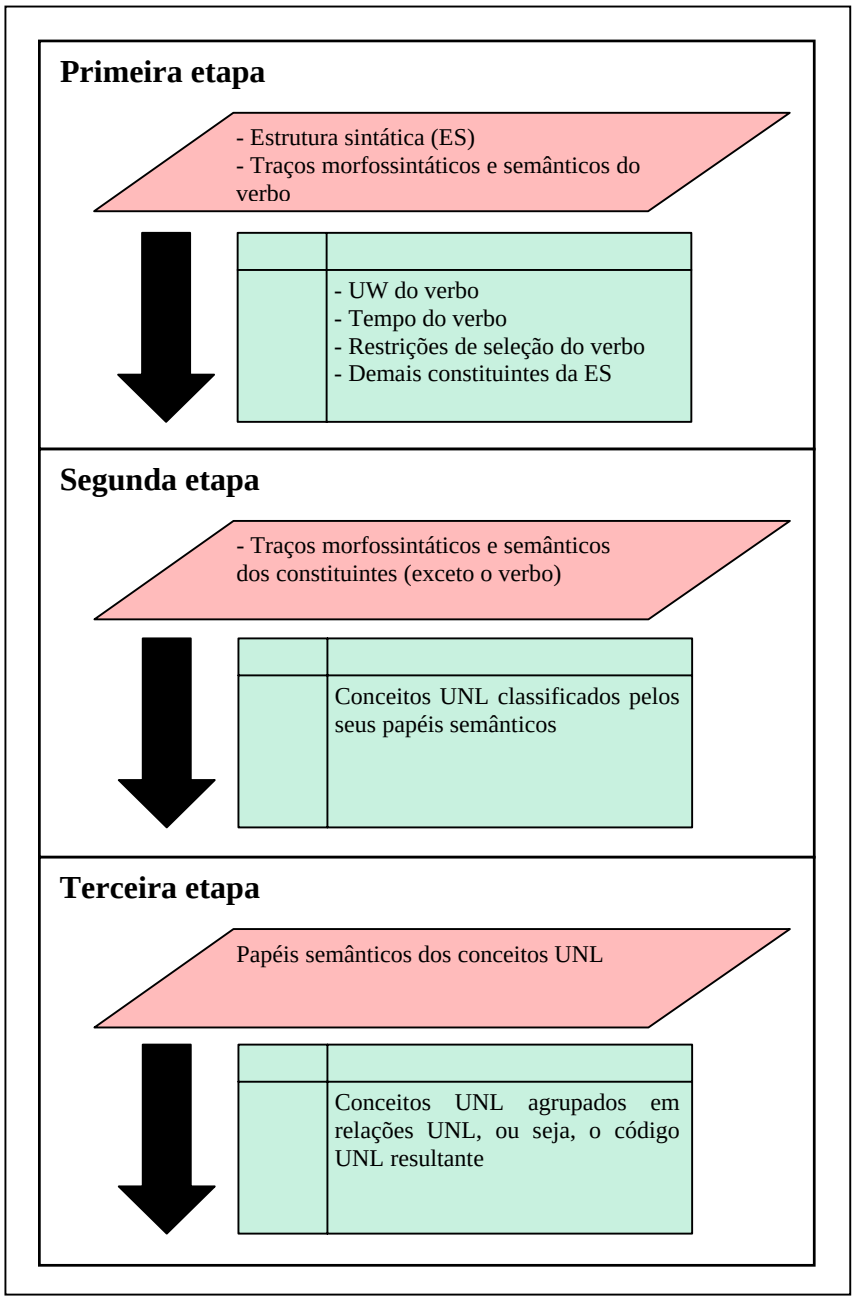


Figura 3 – As três etapas do processo de mapeamento

Os atributos morfossintáticos que estabelecem restrições definem o ambiente sintático da sentença sendo mapeada. Esses atributos incluem a estrutura sintática e alguns traços morfossintáticos lexicais dos constituintes, como a transitividade do verbo.

Da mesma forma, os atributos semânticos que estabelecem restrições, quais sejam, a classe do verbo e a sua subcategorização, juntamente com os atributos de correspondência e os atributos relacionais, que conduzem o mapeamento para diferentes estruturas conceituais, definem o ambiente semântico dessa sentença.

Cada *template*, portanto, representa um ambiente sintático e um ambiente semântico específicos. Variações nos atributos restritivos implicam a necessidade de novos *templates*. As sentenças (3), por exemplo, não apresentam variações em nenhum dos ambientes e são, portanto, representadas pelo mesmo *template*.

(3) a. Considere seu desgaste físico.
b. Considere seu desconforto espiritual.

Da mesma forma, as sentenças (4), que possuem valores diferentes apenas para um atributo morfossintático não restritivo, a saber, o número do substantivo, são representadas por um único *template*.

(4) a. A lua transita este signo.
b. Os astros transitam este signo.

Já as sentenças (5) diferem em um atributo morfossintático restritivo, a estrutura sintática e, portanto, têm diferentes ambientes sintáticos e são mapeadas por dois *templates* diferentes.

(5) a. Hoje o Sol em Sagitário iluminará seu instinto.
b. O Sol em Sagitário iluminará seu instinto hoje.

As sentenças (6), por sua vez, apesar de possuírem o mesmo ambiente sintático, têm diferentes representações conceituais, que são geradas por diferentes atributos de correspondência e atributos relacionais, portanto, os ambientes semânticos são diferentes. Assim, são necessários dois *templates* para mapear essas sentenças, um para mapear advérbios no papel semântico "lugar" (6a) e outro para mapear advérbios no papel semântico "tempo" (6b).

(6) a. O sol brilha em sagitário.
b. O sol brilha em janeiro.

Para modelar o mapeamento de acordo com essas convenções, foram criados 33 *templates* para as 36 sentenças do *corpus* base, um para cada ambiente sintático e semântico. Esse número coincide com o número de estruturas sintáticas diferentes desse corpus, uma vez que essas estruturas representam a única variação morfossintática restritiva nos ambientes sintáticos e as sentenças que têm estrutura sintática idêntica ocorrem no mesmo ambiente semântico sendo, portanto, contempladas pelo mesmo *template*. A seguir, ilustramos a utilização desses *templates* nas três etapas do processo de mapeamento.

4.2 A primeira etapa do mapeamento

A primeira etapa do mapeamento utiliza a **estrutura sintática** da sentença e alguns atributos morfossintáticos e semânticos do **verbo** como restrições do *template*. A estrutura sintática, conforme ilustrado na Figura 2, apresenta as funções sintáticas e categorias sintáticas dos constituintes da sentença, assim como os constituintes, propriamente ditos. A unificação da estrutura sintática do *template* com a estrutura sintática da sentença só ocorre quando elas são completamente idênticas, exceto pelas palavras da sentença, que são substituídas por variáveis, de modo que a mesma estrutura sintática possa contemplar sentenças de forma idêntica, mas com palavras diferentes.

Alguns dos constituintes da estrutura sintática não são relevantes ao mapeamento, portanto, não são repassados como resultado para a segunda etapa. Isso ocorre principalmente com algumas preposições, por exemplo, a preposição "de" em complementos nominais, como ocorre na sentença (7). Nesses casos, aquela preposição (ou suas formas derivadas) é a única que pode manter a função sintática da construção (no exemplo, "de sua perna" como de

complemento nominal), pois se for substituída por outra preposição a sentença se torna não-gramatical ou obtém-se uma estrutura sintática diferente.

(7) Você sabe o tamanho de sua perna.

Em outros casos, no entanto, a preposição é relevante ao mapeamento e precisa ser mantida como uma restrição, como ocorre na sentença (8). Nessa sentença, a preposição "em", se substituída por "de", por exemplo, mantém a função sintática da construção "em sagitário" como complemento nominal, mas altera a função semântica dessa construção, pois na sentença (8) "em" indica "lugar", enquanto "de" poderia indicar "modo" ou "possessão".

(8) Hoje o sol em sagitário iluminará seu instinto

Os atributos do verbo utilizados nessa primeira etapa representam informações lexicais tanto sobre a sua forma analisada quanto sobre a sua forma canônica[7], conforme mostra a Tabela 4. Os dois tipos de informação são importantes para o mapeamento, pois alguns atributos são representados somente na descrição lexical da forma analisada do verbo, enquanto outros, na descrição lexical da forma canônica.

Tabela 4 – Atributos dos verbos na primeira etapa do mapeamento

Atributo	**Exemplo**	**Utilização**
tempo/modo da forma analisada	imperativo afirmativo, futuro do presente, futuro do pretérito	como resultado para a segunda etapa, pois será mapeado em um AL da UNL associado ao verbo ou ao núcleo do predicado para verbos de ligação
tempo/modo da forma canônica	sempre infinitivo pessoal	como restrição, nas locuções verbais, único caso em que o verbo principal é usado na sentença na sua forma canônica
tipo do verbo na forma canônica	transitivo direto, pronominal, bitransitivo, etc.	como restrição morfossintática no *template*
classe do verbo na forma canônica	ação, estado, processo, ação-processo.	como restrição semântica no *template*
subcategorização do verbo na forma canônica	sujeito e objeto, objeto, sujeito e complemento, etc.	como restrição semântica no *template*, indicando os argumentos que o verbo exige
restrições de seleção para todos os argumentos obrigatórios do verbo na forma canônica	sujeito → animado, complemento → lugar	como resultado para a segunda etapa, na forma dos traços semânticos que os argumentos do verbo devem apresentar
UW	considere → *consider*	como resultado para a segunda etapa, pois será representado na codificação UNL final

Em suma, que os valores dos atributos restritivos do verbo do *template* devem ser unificados com as informações da descrição lexical do(s) verbo(s) da sentença, enquanto os

[7] O Léxico Enriquecido, que armazena as informações lexicais para o mapeamento, é composto por dois tipos entradas: formas canônicas e formas analisadas das palavras das sentenças do *corpus*. As formas canônicas são as formas básicas das palavras. As formas analisadas são as variantes de uma forma canônica, incluindo, por exemplo, flexões de gênero, número, grau, modo e tempo.

demais atributos do verbo são transferidos para a etapa seguinte do mapeamento. Em determinados *templates*, alguns atributos do verbo não relevantes para a UNL não são verificados ou repassados para a segunda etapa. Por exemplo, quando o verbo é de ligação, geralmente não é representado na UNL, portanto, não é necessário o atributo "UW".

Nessa primeira etapa, primeiramente o sistema procura unificar a estrutura sintática da sentença de entrada com a de um dos *templates* disponíveis. Quando isso ocorre, o sistema passa então a preencher os traços do verbo dessa estrutura com informações recuperadas do Léxico Enriquecido. Caso esta última unificação não ocorra, o sistema abandona aquele *template* e tenta encontrar outro. Caso ela ocorra com sucesso, o sistema instancia os atributos de resultado e passa para a segunda etapa do processo de mapeamento.

Os seguintes atributos instanciados são repassados para a segunda etapa: 1) UW do verbo, quando o verbo tiver significado próprio (verbos de ligação geralmente não tem); 2) demais constituintes relevantes, no modo em que aparecem na sentença: em português e na sua forma analisada; 3) tempo do verbo; e 4) restrições de seleção do verbo para seus diferentes argumentos, quando existirem.

Por exemplo, para a sentença (7), aqui repetida, a primeira etapa retorna os atributos, devidamente instanciados, da Tabela 5.

(7) Você sabe o tamanho de sua perna.

Tabela 5 – Resultado da primeira etapa do mapeamento da sentença (7)

Atributo	**Valor**
UW do verbo	*know*
restrições de seleção do objeto	não humano
tempo do verbo	presente
pronome 1	você
artigo	o
substantivo 1	tamanho
pronome 2	sua
substantivo 2	perna

4.3 A segunda etapa do mapeamento

Na segunda etapa, o sistema procura atribuir um papel semântico a cada constituinte sintático retornado pela primeira etapa do mapeamento e identificar o seu correspondente conceito UNL, já associado aos seus devidos ALs.

4.3.1 Atribuição dos papéis semânticos e identificação dos conceitos

Para a atribuição dos papéis semânticos e identificação dos conceitos, novamente, os *templates* representam alguns atributos morfossintáticos e semânticos como restrições. Os atributos morfossintáticos utilizados dependem da categoria sintática das palavras que compõem o constituinte. Por exemplo, um constituinte formado por substantivo e artigo apresenta atributos morfossintáticos para esses dois elementos. Os atributos semânticos consistem das restrições de seleção do verbo. Por exemplo, o verbo "saber", na sentença (7), apresenta uma restrição de seleção para o seu objeto (não humano), enquanto o verbo "considerar" apresenta restrições de seleção para o seu sujeito (animado) e para o seu objeto (abstrato).

Os constituintes da sentença (7), por exemplo, devem ser mapeados nos papéis semânticos ilustrados na segunda coluna da Tabela 6, desde que satisfaçam as restrições estabelecidas por esses papéis, como indica a terceira coluna dessa tabela.

Tabela 6 – Atribuição de papéis semânticos aos constituintes da sentença (7)

Constituinte sintático	Papel semântico	Restrições
pronome 1 (você)	experimentador	o constituinte deve ser um pronome de tratamento
UW do verbo (know)	evento	-
artigo (o) + substantivo 1 (tamanho)	objeto	o constituinte deve ser composto por um artigo e um substantivo, sendo que os traços semânticos desse substantivo devem satisfazer as restrições de seleção do verbo da sentença (não humano)
pronome 2 (sua)	possuidor	o constituinte deve ser um pronome possessivo
substantivo 2 (perna)	modificador	o constituinte deve ser um substantivo

Nessa mesma fase, além de determinar o papel semântico dos constituintes, são identificados os conceitos UNL desses constituintes. O conjunto de conceitos, associados aos seus papéis semânticos representam, então, o resultado da segunda etapa do mapeamento, como indica a Tabela 7 para a sentença (7), por exemplo.

Tabela 7 – Conceitos de cada constituinte da sentença (7)

Conceito UNL	Papel semântico
you	experimentador
know.@entry	evento
size.@def	objeto
you	possuidor
leg	modificador

Identificação dos papéis semânticos

Na codificação prévia manual das sentenças, a definição dos papéis semânticos que deveriam ser atribuídos aos constituintes das sentenças do *corpus* foi realizada de forma empírica, com base em três critérios: 1) na especificação da linguagem UNL; 2) na classificação e descrição dos verbos do dicionário de Borba (1990); e 3) no estudo realizado por pesquisadores do NILC sobre as manifestações morfossintáticas da UNL no Português (Sossolote et al., 1997).

A especificação UNL, ao descrever quais as possíveis relações dessa linguagem, define, em alguns casos, quais os possíveis papéis semânticos esperados para os conceitos que fazem parte da relação. Nesses casos, tal definição nos possibilitou uma indicação de uma possível correspondência. Por exemplo, a relação *agt* (agente) define um "agente" que inicia uma ação.

O dicionário de verbos de Borba (1990) foi utilizado na criação do Léxico Enriquecido para fornecer informações semânticas dos constituintes, a saber, a classe de cada verbo, sua subcategorização, suas restrições de seleção e os traços semânticos dos substantivos. No dicionário, as restrições de seleção dos verbos, em alguns casos, são dadas em função do

papel semântico dos constituintes, o que nos forneceu uma indicação da correspondência sintático-conceitual dos constituintes. Por exemplo, para o verbo "ingressar", Borba identifica como restrições de seleção a necessidade de um sujeito "agente" e de um complemento "locativo".

O estudo das manifestações morfossintáticas da UNL no português foi feito pelo NILC durante a criação do módulo de decodificação UNL-Português, visando identificar as possíveis construções gramaticais correspondentes a todas as estruturas UNL de um determinado *corpus*. De acordo com o que foi levantado nesse estudo, uma forma de manifestação morfossintática da UNL no português é justamente aquela decorrente das suas relações semânticas. Tal estudo foi baseado na identificação das possíveis realizações sintáticas dos relacionamentos semânticos entre dois conceitos específicos, isto é, para cada função sintática distinta, foram apresentadas as possíveis relações e suas correspondentes realizações lingüísticas manifestadas no *corpus* em questão.

Por exemplo, a função sintática de "sujeito", segundo tal estudo, pode manifestar-se na UNL pelas relações "obj", "agt", "ins", "met", "cau" e "soj"[8]. Para o caso específico da correspondência entre "sujeito" e "ins", as seguintes informações foram apresentadas (Sossolote et al., 1997, p. 27):

> ***ins & sujeito***
> • *ins* se manifesta sintaticamente como sujeito entre um substantivo e um verbo, como em:
> ins(supply, computer_translation) >> ins(fornecer, tradução_automática) >> sujeito (fornecer, tradução_automática)
> Exemplo: *A tradução automática é incapaz de fornecer um resultado de alta qualidade.*

Nesse caso, consideramos que o papel semântico do sujeito é "instrumento". Com base nessas três fontes de informação, os papéis semânticos levantados para as sentenças do *corpus* são listados na Tabela 8.

Tabela 8 – Possíveis papéis semânticos para os constituintes das sentenças do *corpus* base

Papel semântico	Descrição
Agente	constituinte que desencadeia uma ação, sendo origem dele e seu controlador
Atributo	constituinte que descreve um estado ou uma coisa[9] que representa um estado
Beneficiario	constituinte sede da transferência de posse, vítima ou destinatário de um benefício
Cenário	constituinte que define um mundo virtual onde um evento ocorre ou um estado é verdadeiro ou uma coisa existe
Coisa	constituinte nominal que está em um estado ou possui um atributo
Co_coisa	constituinte nominal secundário (não em foco) que está em um estado ou possui um atributo
Coordenador	constituinte que define uma relação conjuntiva entre dois outros constituintes
Destino	constituinte que define o destino de um evento
Evento	constituinte que designa o evento da sentença, quando houver
Experimentador	constituinte que designa uma experiência ligada a uma disposição mental, uma sensação, uma emoção, uma cognição
Lugar	constituinte que define o lugar no qual um evento ocorre, no qual um estado é verdadeiro ou no qual uma coisa existe

[8] As relações "cau" e "soj" não são mais consideradas na especificação UNL atual.
[9] Adotamos o termo "coisa" como uma tradução do termo *thing* utilizado na especificação UNL original, para indicar um constituinte genérico, sem papel semântico específico.

Maneira	constituinte que define a maneira como um evento é realizado ou as características de um estado
Meta	constituinte que identifica o estado final de um objeto, associado ou não a um evento
Modificador	constituinte que restringe a descrição de uma coisa em foco
Objeto	constituinte que é diretamente afetado por um evento ou estado
Origem	constituinte que define a origem de uma coisa
Paciente	constituinte afetado por aquilo que o verbo expressa, que sofre uma mudança de estado, condição ou posição
Parceiro	constituinte que, juntamente com o agente, inicia um evento
Possuidor	constituinte que possui uma coisa
Razão	constituinte que define a razão pela qual um evento ou estado ocorre
Tempo	constituinte que define o momento em que um evento ocorre ou em que um estado é verdadeiro

Um dado papel semântico pode ser atribuído a constituintes de diferentes categorias sintáticas, compostos por diferentes quantidades de palavras. Por exemplo, o papel "sujeito" pode ser atribuído a um substantivo ("marte"), a um artigo e um substantivo ("a lua") ou a um pronome de tratamento ("você").

4.3.2 Atribuição dos rótulos de atributos

Os conceitos identificados nessa etapa do mapeamento, além da UW, propriamente dita, podem ser constituídos por certos rótulos de atributos (ALs). A identificação desses ALs foi realizada durante a identificação dos papéis semânticos dos constituintes e identificacao das suas respectivas UWs. Por exemplo, na atribuição do papel semântico dos constituintes nominais, como aqueles que exercem a função sintática de sujeito ou objeto, verificamos a existência de um artigo associado ao núcleo do constituinte, como em "o tamanho" e, em caso positivo, o tipo do artigo (definido ou indefinido) é mapeado em um AL (@def ou @indef) que é associado ao conceito para indicar se este é um conceito já referenciado ou não. A Tabela 9 exibe uma lista dos ALs contemplados pelo ConPor.

Tabela 9 – Atributos UNL contemplados pelo ConPor

AL	**Descrição**
@pl	conceito coletivo
@def	conceito já referenciado
@indef	conceito de uma classe não específica
@future	evento ocorrerá no futuro
@past	evento ocorreu no passado
@imperative	evento imperativo
@not	complemento do evento
@possibility	possibilidade lógica de que um evento ocorra
@entry	conceito principal da sentença ou do escopo

Os ALs utilizados no ConPor representam um subconjunto do total de ALs da UNL[10], incluindo alguns provenientes de traços morfossintáticos dos constituintes, recuperados do Léxico Enriquecido, e outros provenientes da estrutura sintática da sentença. Alguns dos ALs disponíveis não são utilizados simplesmente por não terem sido identificados no *corpus*.

[10] A especificação da linguagem UNL dispõe de cerca de 60 ALs.

Outros, no entanto, exigem conhecimento mais profundo para a sua identificação (extralingüístico), como é caso de determinados ALs de aspecto e ponto de vista (@admire, @conclusion, @induce, @begin-just, @begin-soon, @complete, etc.) e não são, portanto, considerados no ConPor, cujo processo de mapeamento se baseia apenas em conhecimento lingüístico. Na verdade, a identificação de alguns desses atributos no Projeto UNL não é prevista de forma automática, mas sim por um processo de pós-edição humana.

Os três primeiros ALs da Tabela 9 são atribuídos a substantivos, sendo o primeiro (@pl) identificado por meio do traço "número" do próprio substantivo e os outros dois (@def ou @indef) por meio do traço "tipo" do artigo que acompanha esse substantivo.

Todos os demais atributos são associados ao evento ou ao estado da sentença. Os três primeiros (@future, @past, @imperative) são identificados a partir do traço modo/tempo do verbo dessa sentença. Os dois seguintes (@not, @possibility) são identificados a partir de constituintes que acompanham o verbo principal (advérbio de negação (não) e verbo auxiliar (poder)). Seguindo a especificação UNL, o AL *@entry* e todos esses demais, nas estruturas com verbos que não são de ligação, foram associados ao conceito do verbo principal; já nas estruturas com verbos de ligação, foram associados ao núcleo do predicado.

Finalizada essa segunda fase, de identificação dos conceitos e seus papéis semânticos, partimos para a identificação de seu inter-relacionamento semântico, conforme será descrito na próxima seção.

4.4 A terceira etapa do mapeamento

A UNL compreende um conjunto de 41 possíveis relações semânticas entre conceitos. No ConPor, a identificação dessas relações ocorre, em parte, implicitamente na segunda etapa, quando são atribuídos os papéis semânticos dos constituintes, contudo, a organização dos conceitos que exercem esses papéis nas relações UNL para cada sentença de entrada só ocorre efetivamente na terceira etapa do mapeamento.

Nessa etapa, com base nos códigos gerados manualmente para as sentenças do *corpus* base, as relações apropriadas para os conceitos UNL resultantes da segunda etapa foram explicitamente indicadas. Por exemplo, o *template* para mapear a sentença (7), aqui repetida, combina os conceitos nas relações UNL indicadas na Tabela 10.

(7) Você sabe o tamanho de sua perna.

Tabela 10 – Relações UNL para os conceitos da sentença (7)

Conceitos	**Relação**
you	aoj(know.@entry, you)
know.@entry	
know.@entry	obj(know.@entry, size.@def)
size.@def	
size.@def	mod(size.@def, leg)
leg	
leg	pos(leg, you)
you	

Na Tabela 11 listamos as 19 relações UNL utilizadas no ConPor. Novamente, essas relações são apenas uma parte do conjunto das relações sentenciais UNL[11], obtidas em

[11] A especificação da linguagem UNL dispõe, atualmente, de 41 relações semânticas.

conseqüência das sentenças do *corpus* base, visto que foram codificadas regras apenas para as relações identificadas nesse *corpus*.

Tabela 11 – Relações UNL contempladas pelo ConPor

RL	**Nome da relação**
agt	Agente
and	Conjunção
aoj	Coisa com atributo
ben	Beneficiário
cao	Co_coisa com atributo
frm	Origem
gol	Meta
man	Maneira
mod	Modificador
obj	Coisa afetada
or	Disjunção
plc	Lugar
plf	Lugar inicial
plt	Lugar final
pos	Possuidor
ptn	Parceiro
rsn	Razão
tim	Tempo
to	Destino

Com a atribuição das relações, finalizamos o processo de mapeamento sintático-conceitual, cujo resultado é um conjunto de relações binárias UNL entre conceitos. Após a modelagem conceitual do processo de mapeamento com base em um corpus de exemplos, partimos para a sua implementação, que será descrita na seção seguinte.

5 A implementação do ConPor

Nessa seção, pretendemos ilustrar o modelo computacional que realiza o mapeamento sintático-conceitual, descrevendo também os subprocessos que tratam dos dados de entrada e de saída e a interface visual que facilita o acesso ao sistema. As linguagens utilizadas para a implementação do sistema foram Prolog (Colmerauer, 1977), para o subprocesso de mapeamento, e Borland Delphi®, para os subprocessos auxiliares e a interface.

A Figura 4 é um detalhamento do módulo **Gerador** da Figura 1. Nela, além dos recursos lingüísticos já descritos, que são utilizados apenas no subprocesso de **mapeamento**, descrito na Seção 4, constam os subprocessos que fornecem suporte a esse mapeamento, na forma de **pré** e **pós-edição**. Esses subprocessos tratam da formatação dos dados de entrada para que estes possam ser manipulados pelo Prolog e dos dados de saída para que estejam de acordo com a sintaxe dos documentos UNL. Nas seções seguintes, descrevemos a implementação desses três subprocessos, bem como da interface de acesso ao sistema.

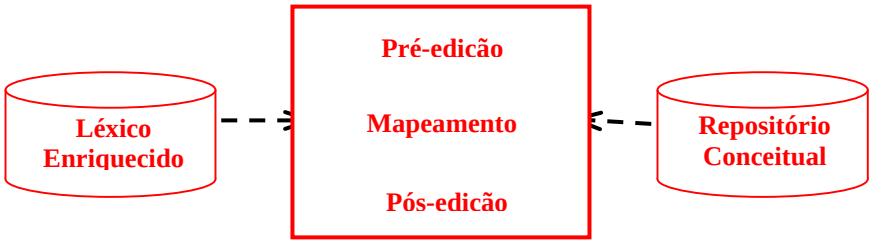


Figura 4 – Subprocessos e recursos do módulo Gerador

5.1 Pré-edição

O formato da estrutura sintática gerada pelo *parser* do NILC não é adequado para manipulação na linguagem Prolog e, portanto, foi alterado por meio de um processo de conversão automática. Esse processo é descrito nos dois passos a seguir, assumindo como entrada a estrutura sintática gerada pelo *parser* para a sentença (8), ilustrada na Figura 5.

(8) Isso acontecerá com alguma pressão.

1) No primeiro passo estão agrupadas todas as funções para a manipulação de *strings* (concatenação, substituição, exclusão, etc.), quais sejam, a eliminação dos caracteres que precedem e sucedem o primeiro casamento (primeira estrutura sintática); a eliminação dos espaços em branco; a eliminação dos retornos de linha; a eliminação dos caracteres entre "#", exceto quando se trata da representação dos terminais; a eliminação das ocorrências da palavra "nucleo"; a inversão da ordem na qual os terminais e suas categorias sintáticas são apresentados; a substituição das ocorrências restantes do caractere "#" por "(" no início da palavra e ")" no fim da palavra; e a conversão de todas as palavras para letras minúsculas. O resultado do primeiro passo é a seguinte estrutura intermediária:

```
(frase(periodo(periodo_independente(suj(suj_simples(sn(pron_subst(isso)))))(predicado(predv(svi(verb
o(acontecerá)))))(aadvo(aadvo_simples(sp(preposicao(com))(sn(aadne(adj(alguma)))(subst(pressão)))))))
)))
```

2) No segundo passo ocorre a transformação efetiva da estrutura sintática em uma lista do Prolog, incluindo colchetes e vírgulas e excluindo parênteses em excesso. O resultado é a estrutura a seguir, utilizada como entrada efetiva para o subprocesso de mapeamento:

```
frase(periodo(periodo_independente([suj(suj_simples(sn(pron_subst(isso)))),predicado(predv(svi(verbo(
acontecerá)))),aadvo(aadvo_simples(sp([preposição(com),sn([aadne(adj(alguma)),subst(pressão)])])))])
)
```

```
(FRASE #Isso acontecerá com alguma pressão#
  (PERIODO #Isso acontecerá com alguma pressão#
    (PERIODO_INDEPENDENTE #Isso acontecerá com alguma pressão#
      (SUJ #Isso#
        (SUJ_SIMPLES #Isso#
          (SN #Isso#
            (nucleo #isso# pron_subst)
          )
        )
      )
      (PREDICADO #acontecerá#
        (PREDV #acontecerá#
          (SVI #acontecerá#
            (nucleo #acontecerá# verbo)
          )
        )
      )
      (AADVO #com alguma pressão#
        (AADVO_SIMPLES #com alguma pressão#
          (SP #com alguma pressão#
            (nucleo #com# preposicao)
            (SN #alguma pressão#
              (AADNE #alguma#
                (nucleo #alguma# adj)
              )
              (nucleo #pressão# subst)
            )
          )
        )
      )
    )
  )
)
```

Figura 5 – Estrutura sintática da sentença (8)

O algoritmo que executa esses dois passos da conversão foi implementado na linguagem Delphi e incorporado à interface de acesso ao sistema.

5.2 Mapeamento

A linguagem Prolog foi utilizada para a especificação dos *templates* de mapeamento principalmente porque possui um mecanismo de inferência próprio, que facilita a unificação dos *templates* com a estrutura sintática de entrada e com as informações das suas entradas lexicais, utilizando, automaticamente, recursos como *backtracking*, quando necessário.

O subprocesso de mapeamento consiste, basicamente, da aplicação dos *templates* Repositório Conceitual, controlada pelo próprio mecanismo de inferência da linguagem Prolog, conforme a estrutura sintática fornecida como entrada. Os *templates* foram especificados de acordo com o formalismo de representação lógica, conforme o exemplo dado a seguir, definido para a primeira etapa do mapeamento. Esse *template*, assim como os demais, além dos atributos citados na Seção 4, possui alguns predicados que representam as operações a serem realizadas sobre os valores dos atributos (tanto aqueles fixos quanto aqueles resultantes da unificação). Fundamentalmente, esses predicados são responsáveis pela manipulação de *strings* e pelas chamadas às etapas posteriores do mapeamento. O *template* do exemplo é utilizado para mapear a estrutura sintática da sentença (8):

```
template(ES,UNL,Sent) :- ftemplate(ES,Pro,V,P,A,S),
                         v(sin(T),can(CanV),[V|_],[]),
                         v(sin(_,int),unl(UnlV),cl(processo),sub([suj]),_,[CanV|_],[]),
                         string_atom(StrPro,Pro),
                         string_atom(StrV,V),
                         string_atom(StrP,P),
                         string_atom(StrA,A),
                         string_atom(StrS,S),
                         stringlist_concat([StrPro,StrV,StrP,StrA,StrS], $ $, Sent),
                         gerarS30(Pro,UnlV,T,A,S,UNL).
```

Cada *template* é representado por um predicado de nome "template", cujos argumentos são a estrutura sintática, como entrada, o código UNL (UNL) e a sentença de entrada (Sent), como saída. Esse predicado remete a vários outros:

- ftemplate(ES,Pro,V,P,A,S) → predicado que verifica a estrutura sintática da sentença, tendo como argumento de entrada essa estrutura (ES) e como argumentos de saída os seus constituintes relevantes (nesse caso, Pro: pronome, V: verbo, P: preposição, A: adjetivo e S: substantivo). No *template* ilustrado acima, o predicado "ftemplate" é instanciado da seguinte forma:

```
ftemplate(frase(periodo(periodo_independente([suj(suj_simples(sn(pron_subst(Pro)
)))],predicado(predv(svi(verbo(V)))),aadvo(aadvo_simples(sp([preposição(P),sn([aa
dne(adj(A)),subst(S)])])))])))),Pro,V,P,A,S).
```

- v(sin(T),can(CanV),[V|_],[]) → predicado que verifica no Léxico Enriquecido os traços da forma analisada do verbo (V), retornando seu tempo (T) e sua canônica (CanV). A descrição da forma analisada do verbo da sentença (8) – "acontecerá" – no Léxico Enriquecido, expressa de acordo com o formalismo da DCG (*Definite Clause Grammar*) (Pereira & Warren, 1980), é a seguinte:

```
v(sin(fut_pres),can(acontecer)) --> [acontecerá].
```

- v(sin(_,int),unl(UnlV),cl(processo),rest(suj(RestSuj)),[CanV|_],[]) → predicado que verifica no Léxico Enriquecido os traços da forma canônica do verbo (CanV), retornando sua UW (UnlV), desde que seja intransitivo (int), que sua classe seja "processo" e que tenha um sujeito como argumento de subcategorização (suj). São recuperadas também as restrições de seleção do verbo, em RestSuj. A descrição da forma canônica do verbo da sentença (8) – "acontecer" – no Léxico Enriquecido é a seguinte:

```
v(sin(inf_pess,int),unl(happen),cl(processo),rest([suj([abstrato])])) --> [acontecer].
```

- string_atom(StrPro,Pro),string_atom(StrV,V),string_atom(StrP,P), string_atom(StrA,A), string_atom(StrS,S) → predicados do Prolog que convertem os termos que representam os constituintes da sentença em *strings*, para que

possam ser manipulados como tal. Assumem como entrada o termo (Pro, V, P, A, S) e retornam como saída a *string* desse termo (StrPro, StrV, StrP, StrA, StrS).

- stringlist_concat([StrPro,StrV,StrP,StrA,StrS], $ $, Sent) → predicado do Prolog que concatena as várias *strings* dos constituintes da sentença (StrPro, StrV, StrP, StrA, StrS), incluindo um caractere separador entre elas (neste caso, um espaço). Retorna a sentença de entrada (Sent).
- gerarS30(Pro,UnlV,T,A,S,UNL) → predicado que dispara o processamento da segunda etapa do mapeamento, passando como argumentos a UW e o tempo do verbo e os demais constituintes relevantes da sentença, e retornando o código UNL daquela sentença. Neste caso, o predicado é instanciado para dar início à segunda etapa conforme segue:

```
gerarS30(isso,happen,fut_pres,alguma,pressão,UNL).
```

A estrutura dos demais *templates* é basicamente a mesma, diferindo apenas nos constituintes retornados por "ftemplate", os traços verificados e retornados para o verbo, o número de constituintes convertidos em *strings* e concatenados para representar a sentença de entrada e os constituintes repassados à segunda etapa por meio do predicado "gerarSx", sendo x o número da(s) sentença(s) do *corpus*.

Na segunda etapa, os constituintes são mapeados em seus conceitos e respectivos papéis semânticos, por meio de chamadas de predicados específicos em "gerarS30", um para cada papel diferente. No caso da sentença (8), as chamadas de predicados utilizadas são as seguintes:

```
gerarS30(Pro,UnlV,T,A,S,UNL) :- objeto(Pro,O),
                                evento(UnlV,T,E),
                                maneira(S,Ma),
                                modificador(A,M),
                                codificarS30(O,E,Ma,M,UNL).
```

Esses predicados têm as seguintes funções:

- objeto(Pro,UnlPro) → verifica os traços do pronome (Pro) no Léxico Enriquecido, restringindo seu tipo a "demonstrativo" (dem) e mapeia tal pronome no papel "objeto", retornando a UW desse pronome (UnlPro). Para tanto, possui a seguinte estrutura:

```
objeto(Pro,UnlPro) :- pron(sin(dem),unl(UnlPro),[Pro|_],[]), !.
```

- evento(UnlV,T,E) → concatena, por meio do predicado do Prolog "strcat", a UW do verbo (UnlV) ao atributo que representado seu tempo (T), e ao atributo (@entry). Retorna a *string* resultante dessa concatenação como UW no papel semântico "evento" (E). Possui a seguinte estrutura:

```
evento(UnlV,T,E) :-  string_atom(StrUnlV,UnlV),
                     string_atom(StrT,T),
                     strcat(StrUnlV,$.@entry.@$,StrE0),
                     strcat(StrE0,StrT,StrE),
                     string_atom(StrE,E).
```

- maneira(S,UnlS) → busca no Léxico Enriquecido e retorna a UW (UnlS) do substantivo (S) que representa a maneira como o evento é realizado, mapeado no papel "maneira". Neste caso, como se trata de um substantivo que já está na sua forma canônica (pressão), o predicado utilizado possui a estrutura dada a seguir:

```
maneira(S,UnlS)  :- s(_,unl(UnlS),_,[S|_],[]), !.
```

Caso o substantivo estivesse na sua forma analisada na sentença, outro predicado seria utilizado, remetendo também à descrição lexical da forma canônica:

```
maneira(S,Ma) :- s(sin(N),can(CanS),[S|_],[]),
                 s(_,unl(UnlS),_,[CanS|_],[]),
                 string_atom(StrUnlS,UnlS),
                 string_atom(StrN,N),
                 strcat(StrUnlS,$.@$,StrMa0),
                 strcat(StrMa0,StrN,StrMa),
                 string_atom(StrMa,Ma),!.
```

- modificador(A,UnlA) → busca no Léxico Enriquecido e retorna a UW (UnlA) do adjetivo (A) que modifica a maneira como o evento é realizado, mapeado no papel "modificador". Novamente, se trata de um adjetivo que já está na sua forma canônica (pressão), portanto, o predicado utilizado possui a estrutura dada a seguir:

```
modificador(A,UnlA) :- adj(unl(UnlA),[A|_],[]), !.
```

- codificarS30(O,E,Ma,M,UNL) → dispara o processamento da terceira etapa do mapeamento, passando como argumentos todos os conceitos dos constituintes da sentença classificados segundo seus papéis semânticos (O: objeto, E: evento, Ma: maneira, M: modo) e retornando o código UNL daquela sentença. Para a sentença (8), o predicado é instanciado para dar início à terceira etapa conforme segue:

```
codificarS30(this,happen.@entry.@fut_pres,pressure,some,UNL).
```

A estrutura dos demais *templates* nessa etapa difere na quantidade e no tipo das chamadas de predicados (e nos seus argumentos) para a atribuição dos papéis semânticos e, consequentemente, nos argumentos repassados à terceira etapa do mapeamento por meio do predicado "codificar".

A terceira etapa combina os vários conceitos nas devidas relações semânticas da UNL, convertendo seus termos em *strings* e concatenando-os entre si e às *strings* das relações. Para o exemplo da sentença (8), os predicados utilizados são os seguintes:

```
codificarS30(O,E,Ma,M,UNL) :- string_atom(FimO,O),
                              string_atom(FimE,E),
                              string_atom(FimMa,Ma),
                              string_atom(FimM,M),
                              strcat($obj($,FimE,Ob1),
                              strcat(Ob1,$,$,Ob2),
                              strcat(Ob2,FimO,Ob3),
                              strcat(Ob3,$)$,Ob4),
                              strcat(Ob4,$|$,ObFim),
                              strcat($man($,FimE,Ma1),
                              strcat(Ma1,$,$,Ma2),
                              strcat(Ma2,FimMa,Ma3),
                              strcat(Ma3,$)$,Ma4),
                              strcat(Ma4,$|$,MaFim),
                              strcat($mod($,FimMa,Mo1),
                              strcat(Mo1,$,$,Mo2),
                              strcat(Mo2,FimM,Mo3),
                              strcat(Mo3,$)$,MoFim),
                              strcat(ObFim,MaFim,Inter),
                              strcat(Inter,MoFim,UNL).
```

Ao final do processamento realizado nessa última etapa do mapeamento, os dois parâmetros retornados como saída, ou seja, o código UNL e a sentença que corresponde à estrutura sintática de entrada, são instanciados com duas *strings* conforme ilustrado a Figura 6. Essas *strings* serão, ainda, manipuladas em outro subprocesso, descrito na próxima seção, para que se apresentem no formato apropriado dos documentos UNL.

```
obj(happen.@entry.@fut_pres,this)|man(happen.@entry.@fut_pres,pressure)|mod(pressure,some)
isso acontecerá com alguma pressão
```

Figura 6 – Resultado do processo de mapeamento para a sentença (8).

5.3 Pós-edição

A estrutura conceitual obtida pelo subprocesso de mapeamento é uma estrutura bastante próxima da UNL, no entanto, algumas alterações ainda precisam ser feitas sobre ela para que esteja completamente de acordo com a sintaxe da UNL. Parte dessas alterações é necessária por conta de certos critérios adotados no processo de mapeamento que visam manter a compatibilidade com os recursos lingüísticos utilizados e tornar a implementação mais genérica. Outras alterações são necessárias apenas para atribuir marcadores padrão dos documentos UNL.

Com relação à compatibilidade com formatos de outros recursos, alguns ALs foram inseridos de acordo com a sua sintaxe no Léxico Enriquecido, que preserva, por sua vez, a notação utilizada para os traços provinda do Léxico do Português (Nunes et al., 1996). Essa notação é, algumas vezes, diferente da utilizada pela UNL.

Para generalizar o processo de atribuição dos ALs, alguns deles que não utilizados na UNL foram inseridos no mapeamento. Por exemplo, na UNL, ALs que indicam o tempo de ocorrência do evento são utilizados somente os tempos “passado” e “futuro”. O ConPor, por sua vez, utiliza ALs para todos os tempos, portanto, é preciso excluir os ALs que indicam

outros tempos. Para realizar essas e outras pequenas alterações, criamos um subprocesso de pós-edição dos dados de saída do subprocesso de mapeamento. Alguns exemplos de regras são ilustrados na Tabela 11.

Tabela 11 – Exemplos de regras de pós-edição

Regra	**Justificativa**
Excluir *@pres*	UNL não considera AL para tempo presente
Substituir *@fut_pres* por *@future*	Notação do traço do Léxico Enriquecido diferente da notação do AL na UNL
Substituir *@pret_perf* por *@past*	Notação do traço do Léxico Enriquecido diferente da notação do AL na UNL
Substituir *@de* por *@def*	Notação do traço do Léxico Enriquecido diferente da notação do AL na UNL
Substituir *@i* por *@indef*	Notação do traço do Léxico Enriquecido diferente da notação do AL na UNL
Substituir *@can* por *@possibility*	Verbo modal "poder" (*can*) implica no AL "possibilidade" (*possibility*) da UNL

Após a aplicação dessas regras, a saída do mapeamento da Figura 6 é convertida para a estrutura da Figura 7.

```
obj(happen.@entry.@future,this)|man(happen.@entry.@future,pressure)|mod(pressure,some)
isso acontecerá com alguma pressão
```

Figura 7 – Resultado da aplicação das regras de pós-edição sobre a estrutura da Figura 6

Além dessas regras, foram definidos procedimentos formatar a saída de acordo com a sintaxe dos documentos UNL, separando as sentenças em linhas e acrescentando marcadores pertinentes a esses documentos, para que o documento gerado pudesse ser utilizado como entrada em um sistema de decodificação do Projeto UNL. Consideramos, para tanto, a sintaxe exigida pela versão 2.5 dos decodificadores desse Projeto. Os marcadores utilizados incluem códigos para início e fim de documento ([D] e [/D]), para início e fim de parágrafo ([P:x] e [/P]), para início e fim de sentença ([S:x] e [/S])[12] e para a própria sentença sendo codificada, em comentário (;).

A aplicação deste último conjunto de operações de pós-edição sobre a estrutura da Figura 7, por exemplo, resulta no código UNL ilustrado na Figura 8.

```
[D]
[P]
[S:1]
; isso acontecerá com alguma pressão
obj(happen.@entry.@future, this)
man(happen.@entry.@future, pressure)
mod(pressure, some)
[/S]
[/P]
[/D]
```

Figura 8 – Documento UNL gerado para a sentença (8)

[12] Aqui, x representa um índice numérico que deve ser diferente para cada sentença e parágrafo.

Neste caso, o documento é composto por uma única sentença. Um exemplo de documento com a codificação de três sentenças do *corpus*, que, hipoteticamente, fazem parte de um mesmo parágrafo, é dado na Figura 9.

Códigos UNL como esses, em um sistema de comunicação multilingual que utilize a UNL como interlíngua, podem servir de entrada para o processo de decodificação e ser, portanto, traduzidos para quaisquer línguas naturais previstas pelo sistema.

```
[D]
[P]
[S:1]
; as perspectivas de retorno a um estado anterior estão mais longínquas
mod(perspective.@entry.@def.@pl, regress)
plt(regress, state.@indef)
mod(state.@indef, previous)
aoj(far-away, perspective.@entry.@def.@pl,)
man(far-away, more)
[/S]

[S:2]
; a lua crescente ocorre amanhã
obj(occur.@entry, moon.@def)
mod (moon.@def, crescent)
tim(occur.@entry, tomorrow)
[/S]

[S:3]
; este é um momento delicado
aoj(moment.@indef, this.@entry)
mod(moment.@indef, delicate)
[/S]
[/P]
[/D]
```

Figura 9 – Exemplo de documento UNL gerado para várias sentenças

5.4 Interface de acesso

Para facilitar a utilização do ConPor, foi implementado, em Delphi, um módulo que faz a interface entre usuário e o subprocesso de mapeamento em Prolog. Esse módulo recebe a entrada do usuário para o sistema, realiza as devidas conversões, quando necessárias, por meio das regras de pré-edição, repassa tal entrada convertida ao Prolog e, depois de computados os resultados, formata esses resultados para a sintaxe dos documentos UNL, por meio das regras de pós-edição, e os exibe para o usuário. Além disso, esse módulo permite inserir ou excluir entradas no Léxico Enriquecido.

Para informar a entrada do sistema, o usuário indica a sentença de entrada, no caso das sentenças já contempladas (ou seja, as sentenças do *corpus* base), ou a estrutura sintática de

entrada, no formato do *parser*[13] ou do Prolog. Essas e as outras funções e informações da interface do sistema são ilustradas na Figura 10 e descritas na Tabela 12.

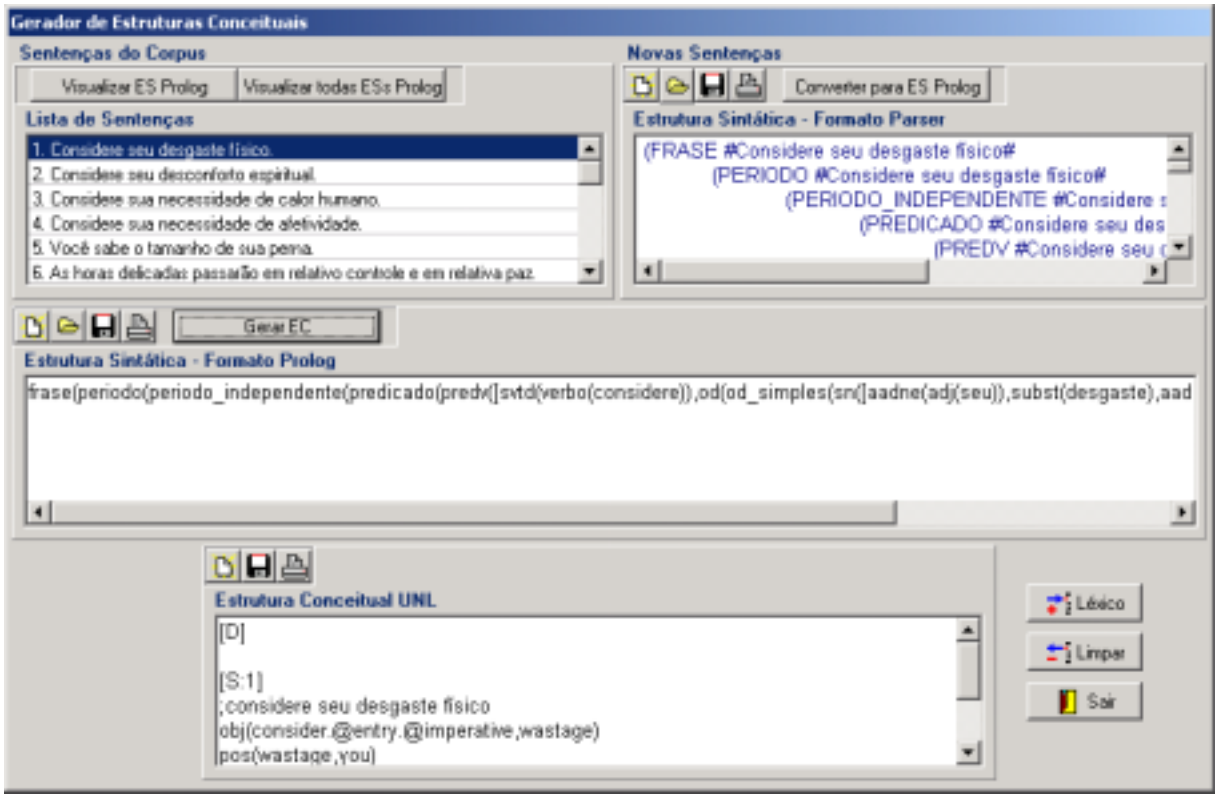


Figura 11 – Tela principal da interface do ConPor

Tabela 12 – Descrição das opções da interface do ConPor

Opção	Descrição
Sentenças do *Corpus* – Lista de Sentenças	Exibe todas as sentenças do *corpus* base
Sentenças do *Corpus* – Visualizar ES Prolog	Exibe a estrutura sintática no formato do Prolog para a sentença selecionada
Sentenças do *Corpus* – Visualizar todas ESs Prolog	Exibe as estruturas sintáticas no formato do Prolog para todas as sentenças do *corpus* base
Novas Sentenças – Estrutura Sintática - Formato *Parser*	Permite informar uma determinada estrutura sintática no formato do *parser*, ou abrir um arquivo texto que a contenha. A estrutura informada pode, ainda, ser editada, salva ou impressa
Novas Sentenças – Converter para ES Prolog	Converte a estrutura sintática no formato do *parser* para seu respectivo formato no Prolog – etapa de pré-edição
Estrutura Sintática – Formato Prolog	Exibe a estrutura sintática no formato do Prolog, seja para a sentença do *corpus* base selecionada ou para a estrutura do *parser* convertida. Permite também abrir um arquivo texto com determinada(s) estrutura(s) já no formato Prolog,

[13] Apesar da arquitetura do sistema prever a escolha da estrutura sintática mais provável, alternativamente, ele pode utilizar como entrada o texto gerado como saída pelo *parser*, inclusive em arquivo (texto), sem seleção manual. Neste caso, a primeira estrutura sintática é automaticamente escolhida.

	editar, salvar e imprimir essa(s) estrutura(s)
Estrutura Sintática – Gerar EC	Dispara o subprocesso de mapeamento do Prolog, que interpreta a(s) estrutura(s) sintática(s) de entrada e retorna como saída o(s) código(s) UNL e a(s) sentença(s) correspondente(s) a essa(s) estrutura(s). Em seguida, aplica automaticamente as regras de pós-edição
Estrutura Conceitual UNL	Exibe os códigos UNL gerados pelo subprocesso de mapeamento e pós-editados. Permite editar, salvar e imprimir os códigos obtidos

A interface de acesso ao Léxico Enriquecido pode ser usada para incluir ou excluir entradas lexicais e é ilustrada na Figura 11. As inserções e a exclusões de entradas lexicais são feitas de acordo com a categoria sintática do item lexical, uma vez que cada categoria apresenta diferentes traços. Por exemplo, para inserir o item “teste”, na sua forma canônica, a primeira guia (Substantivos) é selecionada, os dados são informados, conforme a Figura 11 e, por fim, é selecionado o botão “Gravar”. A entrada inserida no Léxico Enriquecido é a mostrada a seguir:

```
s(sin(singular),unl(test),sem([abstrato,inanimado,nao_humano])) --> [teste].
```

Alterações no Léxico Enriquecido são necessárias quando as palavras da estrutura sintática sendo mapeada pelo sistema não estiverem armazenadas nesse léxico. Considerando que o sistema permite diferentes sentenças de entrada, desde que a estrutura sintática seja contemplada pelos *templates*, essa interface se mostra de grande utilidade, pois, se não fosse definida, o usuário precisaria alterar o código do sistema no Prolog para inserir ou excluir entradas para que pudesse gerar estruturas conceituais para novas sentenças.

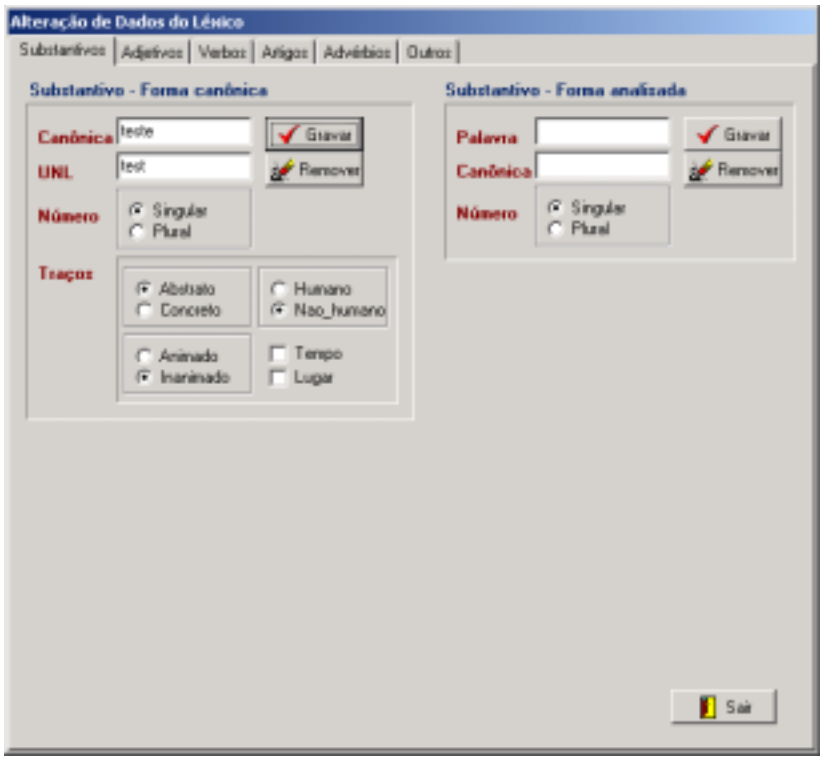


Figura 11 – Tela da interface de acesso ao Léxico Enriquecido

6 Considerações finais

Neste relatório descrevemos o desenvolvimento do sistema ConPor, responsável pelo mapeamento de estruturas sintáticas em estruturas conceituais UNL. Nesse sistema, a interpretação semântica para a obtenção das estruturas conceituais é realizada como uma etapa de processamento posterior ao processamento sintático, utilizando o resultado desse processamento, ou seja, estruturas sintáticas, como entrada. Essa abordagem é adotada em vários outros sistemas, como o *Princitran* (Dorr e Voss, 1996), de tradução automática por interlíngua. Apesar de não aproveitar as possíveis contribuições da intercalação de diferentes etapas, como a influência de informações semânticas na etapa de processamento sintático, tal abordagem se mostrou viável porque a arquitetura do sistema prevê a seleção humana da estrutura sintática mais provável para cada sentença, dentre todas as geradas pelo *parser*. Com isso, foi possível utilizar o *parser* do NILC sem a necessidade de alterá-lo, o que demandaria um trabalho excessivo, considerando sua complexidade, dada em função da sua grande abrangência.

O subprocesso de mapeamento do ConPor foi implementado seguindo o formalismo de representação lógica, por meio de modelos de estruturas (*templates*) que permitiram generalizar o mapeamento de sentenças com a mesma estrutura sintática e mesma representação conceitual. A linguagem utilizada – Prolog – se mostrou adequada para essa tarefa por possuir um mecanismo de inferência próprio para realizar a unificação das estruturas sintáticas de entrada com os *templates* disponíveis no sistema. Alguns ajustes necessários aos dados de entrada e também sobre os dados de saída foram realizados de modo bastante simples, por meio de subprocessos de pré e pós-edição. Esses subprocessos foram

implementados em Delphi juntamente com um módulo que faz a interface entre o usuário e o subprocesso de mapeamento no Prolog, criado com o intuito de facilitar a utilização do sistema.

O sistema descrito aqui representa a primeira versão de um protótipo que deve ser estendido com relação tanto ao número de estruturas sintáticas (e sentenças) previstas no *corpus* base quanto à complexidade gramatical dessas estruturas. Além dessa extensão, deverá ser criado um *corpus* de testes, com sentenças e seus respectivos códigos UNL, para avaliar a qualidade das estruturas conceituais geradas, numa abordagem comparativa. Outras formas de avaliação estão previstas, incluindo a decodificação manual dos códigos gerados e a posterior análise da proximidade semântica entre as sentenças obtidas e as originais.

Referências bibliográficas

Borba, F.S. (1990). *Dicionário gramatical de verbos do português contemporâneo do Brasil*. Fundação Editora Unesp, São Paulo.

Dias-da-Silva, B.C.; Sossolote, C.; Zavaglia, C.; Montilha, G.; Rino, L.H.M.; Nunes, M.G.V.; Oliveira Jr., O.N.; Aluísio, S.M. (1998). The design of the Brazilian Portuguese machine tractable dictionary for an interlíngua sentence generator. In *III Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada*. PUCRS, Porto Alegre.

Dorr, B.J. (1992). The Use of Lexical Semantics in Interlingual Machine Translation. *Journal of Machine Translation*, 7:3, pp. 135-193.

Dorr, B.J. (1993). *Machine Translation: A View from the Lexicon*. MIT Press, Cambridge.

Door, B.J.; Voss, C. (1996). A Multi-Level Approach to Interlingual MT: Defining the Interface between Representational Languages. In L.M. Iwanska and S.C. Shapiro (ed.), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. The MIT Press, 2000.

Fillmore, C. (1968). The case for case. In E. Bach and R.T. Harms (orgs.), *Universals in linguistic theory*, pp. 1-88. Rinehard and Winston, New York.

Jackendoff, R. (1990). *Semantic Structures*. The MIT Press, Cambridge.

Kaplan, R.M.; Bresnan, J. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*. The MIT Press, Cambridge.

Minsky, M. (1975). A Framework for Representing Knowledge. In P. Wiston (ed.), *The Psychology of Computer Vision*. McGraw-Hill, New York.

Nunes, M.G.V.; Vieira, F.M.C.; Zavaglia, C.; Sossolote, C.R.C.; Hernandez, J. (1996). *A Construção de um Léxico da Língua Portuguesa do Brasil para suporte à Correção Automática de Textos*. Série de Relatórios Técnicos do ICMC, (Tech. Rep. 42), Universidade de São Paulo, São Carlos.

Pereira, F.C.N.; Warren, D.H.D. (1980). Definite Clause Grammars for Language Analysis – A Survey of the Formalism and a Comparison with Augmented Transition Networks. In *Artificial Intelligence*, 13, pp. 231-278.

Schank, R. (1975). *Conceptual Information Processing*. North-Holland Publishing Company, New York.

Sossolote, E.R.C.; Zavaglia, C.; Rino, L.H.M.; Nunes, M.G.V. (1997). *As Manifestações Morfossintáticas da Linguagem UNL no Português do Brasil*. Série de Relatórios Técnicos do NILC, NILC-TR-97-02 (Notas do ICMC, no.36). São Carlos, Novembro, 76p.

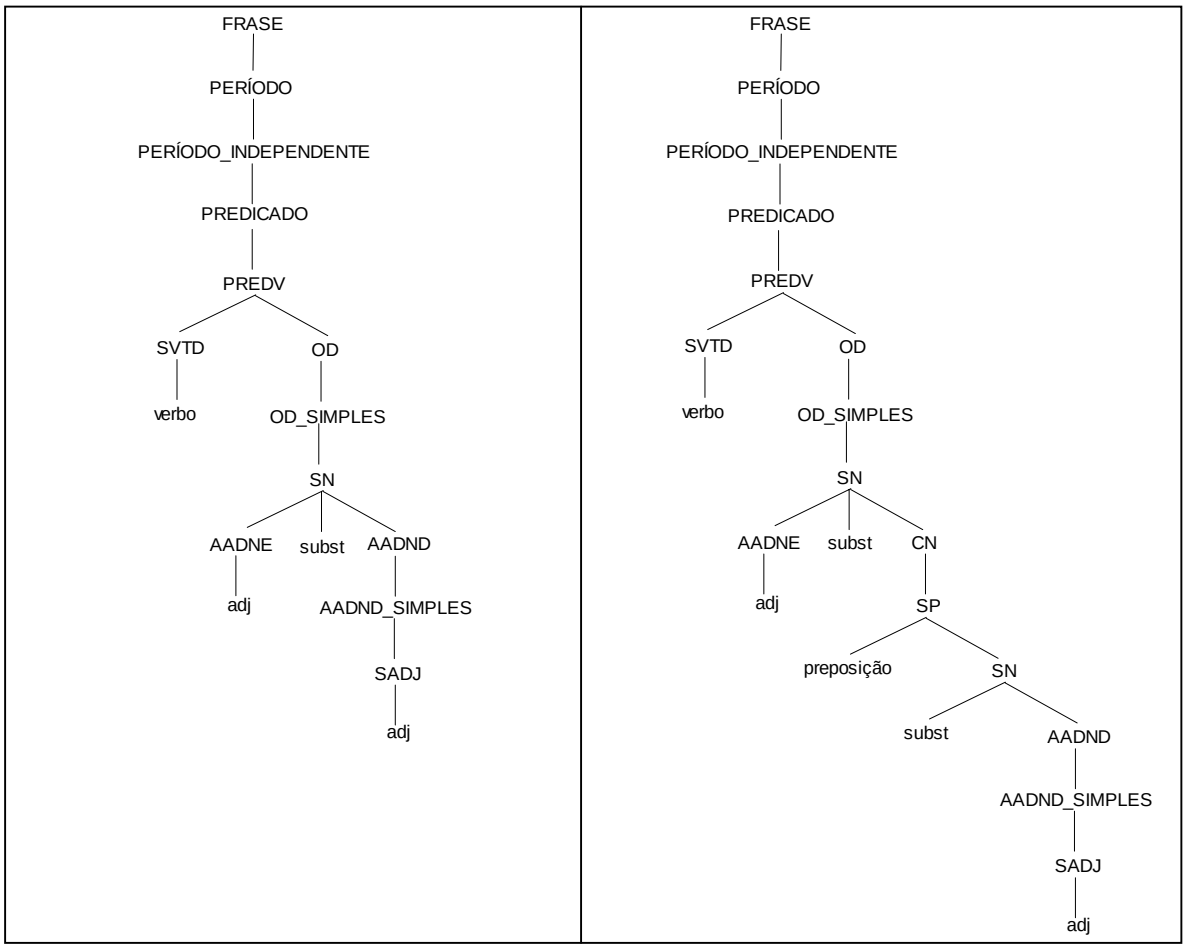
Specia, L.; Rino, L.H.M. (2002). *O desenvolvimento de um léxico para a geração de estruturas conceituais UNL*. Série de Relatórios Técnicos do NILC, NILC-TR-02-14. São Carlos, Setembro, 25p.

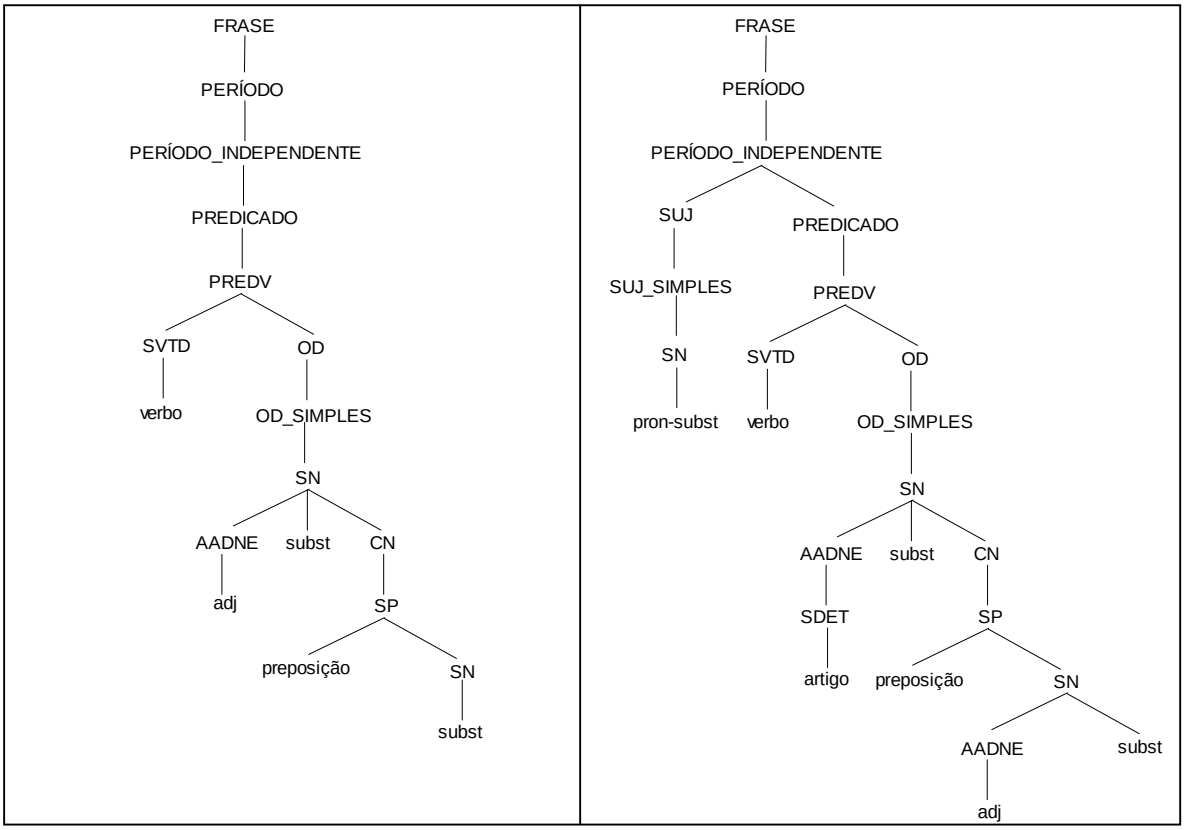
Uchida, H.; Zhu, M.; Senta, T.D. (1999). *The UNL, a Gift for a Millennium*. UNU/IAS/UNL Center, Tokyo.

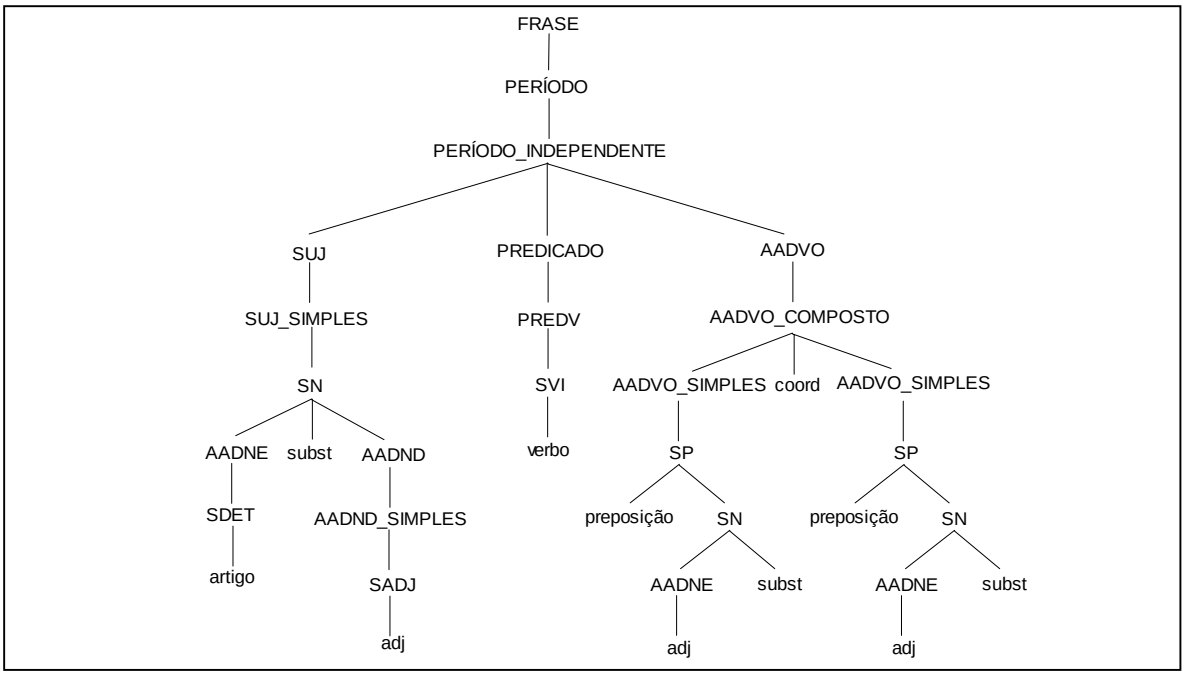
UNL (2001). *The Universal Networking Language (UNL) Specifications*. UNU/IAS/UNL Center, Tokyo.

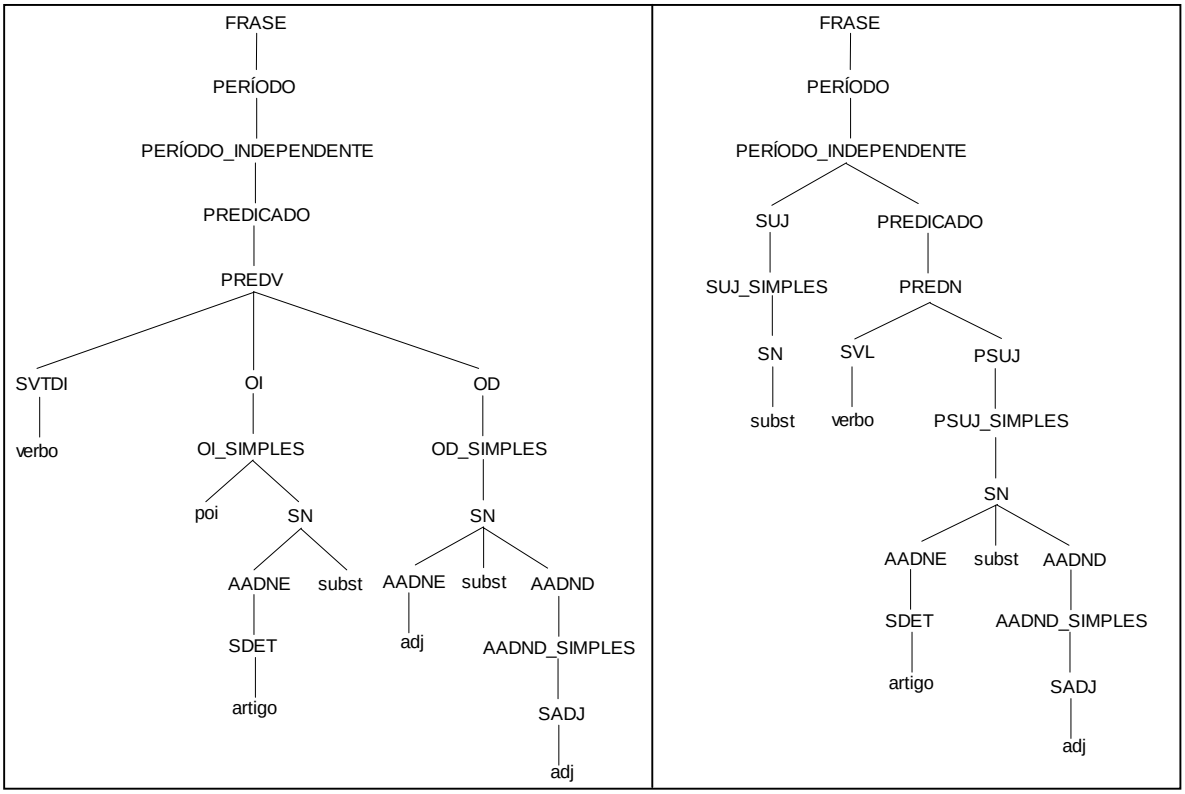
Apêndice A

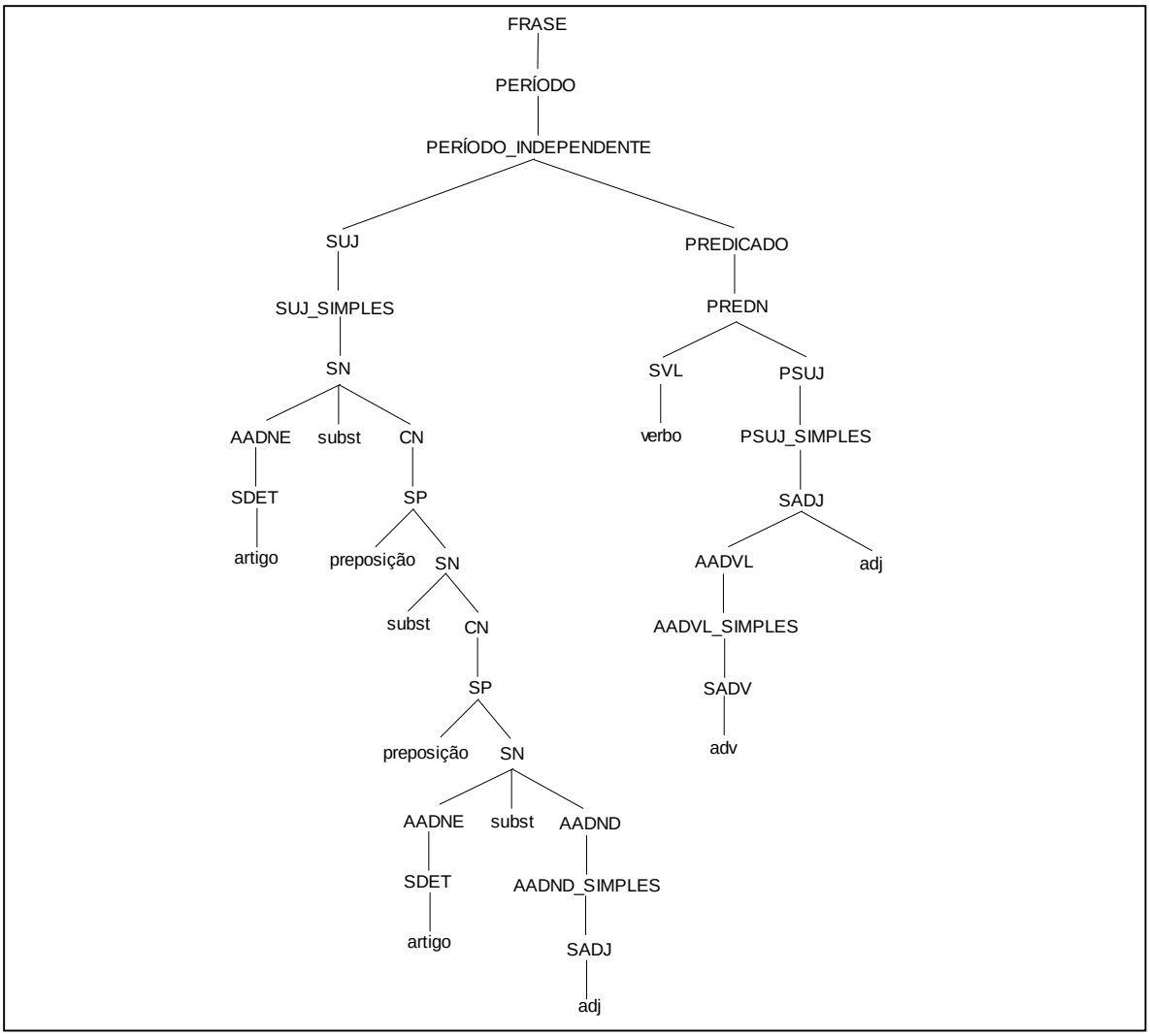
A seguir, apresentamos as 33 estruturas sintáticas contempladas pelo *corpus* base do ConPor, numa representação em árvore de derivação, preservando a notação dos símbolos terminais e não-terminais das regras gramaticais do *parser* do NILC aplicadas para obter tais estruturas.

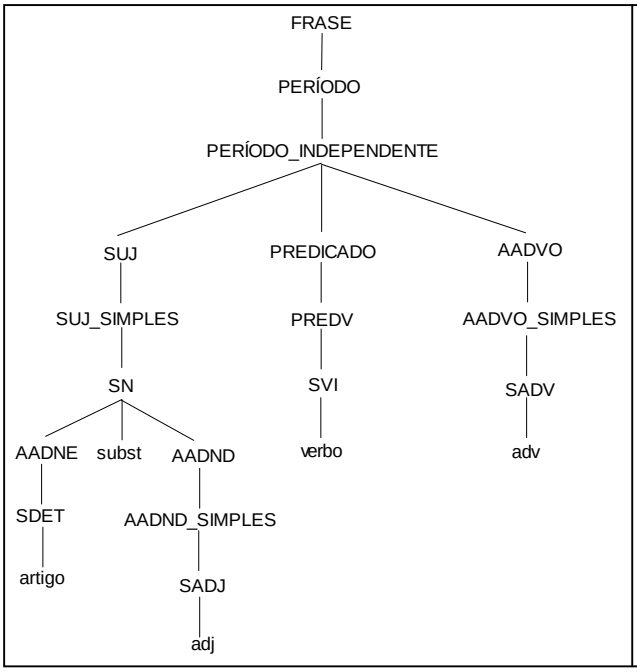


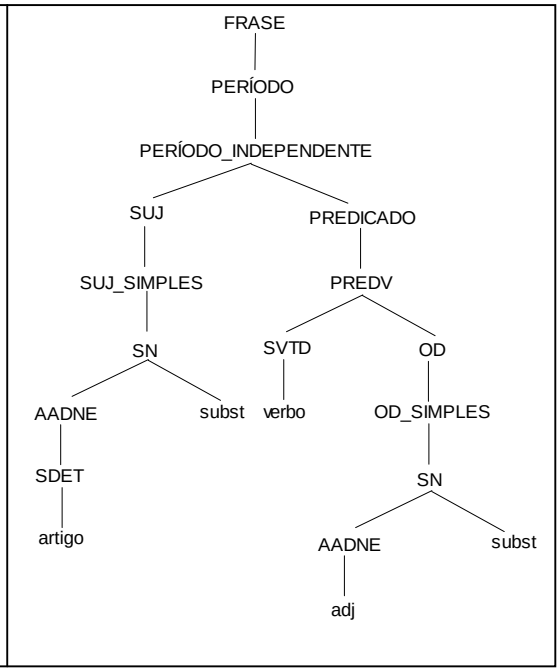

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTD
OD
verbo
OD_SIMPLES
SN
AADNE
subst
CN
adj
SP
preposição
SN
subst
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTD
OD
pron-subst
verbo
OD_SIMPLES
SN
AADNE
subst
CN
SDET
SP
artigo
preposição
SN
AADNE
subst
adj

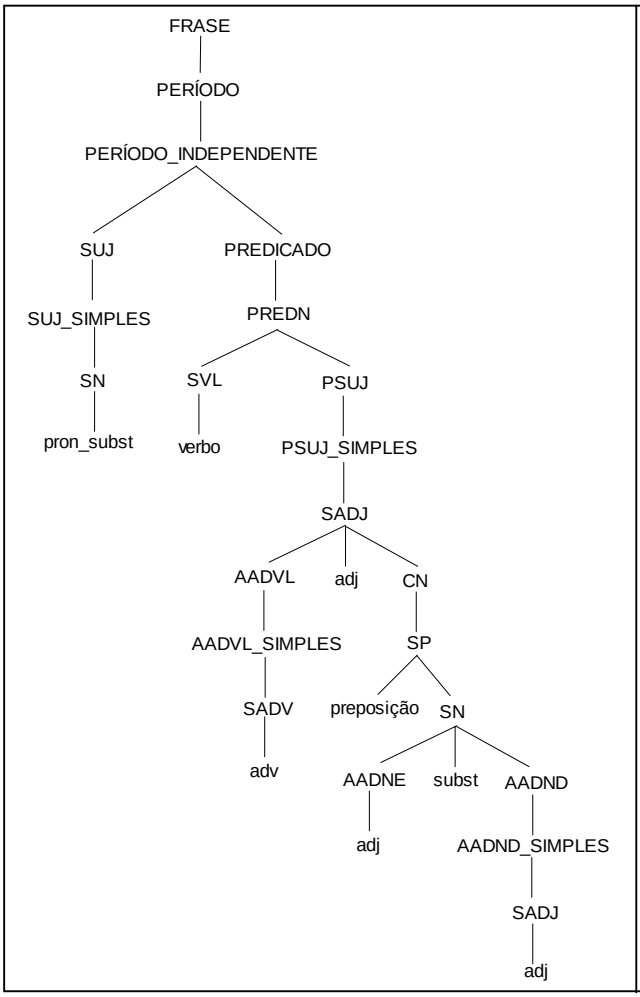

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
AADVO
SUJ_SIMPLES
PREDV
AADVO_COMPOSTO
SN
SVI
AADVO_SIMPLES
coord
AADVO_SIMPLES
AADNE
subst
AADND
verbo
SP
SP
SDET
AADND_SIMPLES
preposição
SN
preposição
SN
artigo
SADJ
AADNE
subst
AADNE
subst
adj
adj

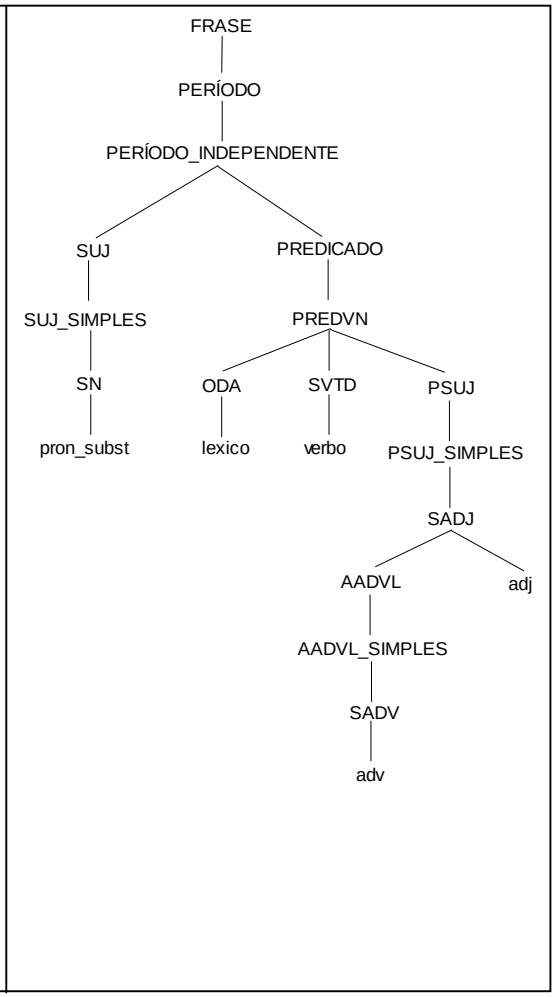

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTDI
verbo
OI
OI_SIMPLES
poi
SN
AADNE
subst
SDET
artigo
OD
OD_SIMPLES
SN
AADNE
subst
adj
AADND
AADND_SIMPLES
SADJ
adj
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
SUJ_SIMPLES
SN
subst
PREDICADO
PREDN
SVL
verbo
PSUJ
PSUJ_SIMPLES
SN
AADNE
subst
SDET
artigo
AADND
AADND_SIMPLES
SADJ
adj

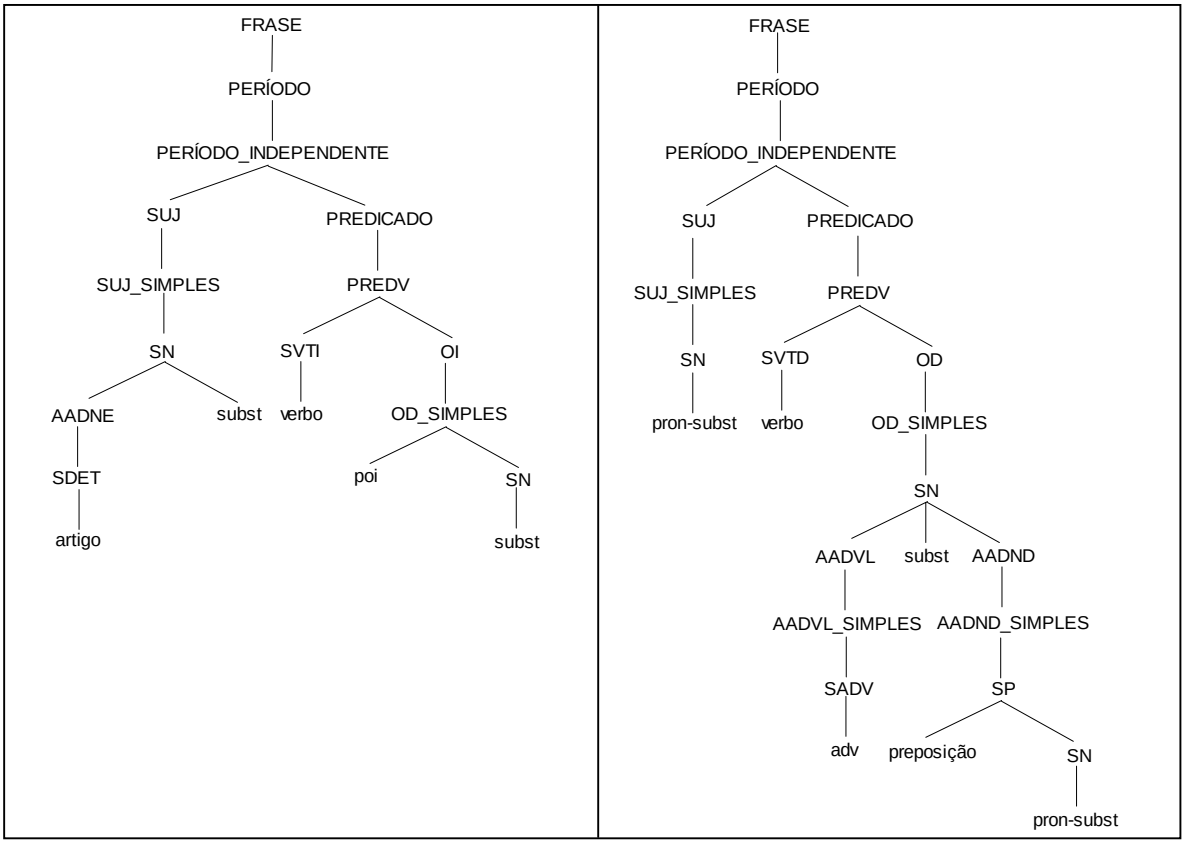

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
SUJ_SIMPLES
SN
AADNE
subst
SDET
artigo
CN
SP
preposição
SN
subst
CN
SP
preposição
SN
AADNE
subst
SDET
artigo
AADND
AADND_SIMPLES
SADJ
adj
PREDICADO
PREDN
SVL
verbo
PSUJ
PSUJ_SIMPLES
SADJ
AADVL
adj
AADVL_SIMPLES
SADV
adv

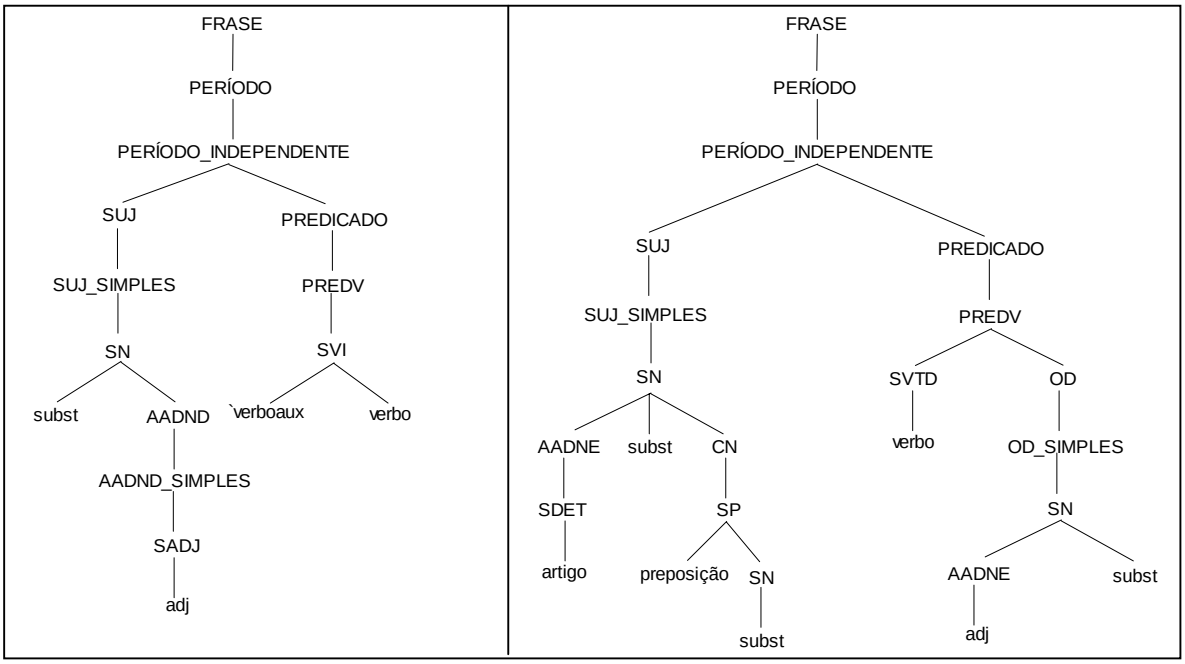

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
AADVO
SUJ_SIMPLES
PREDV
AADVO_SIMPLES
SN
SVI
SADV
AADNE
subst
AADND
verbo
adv
SDET
AADND_SIMPLES
artigo
SADJ
adj

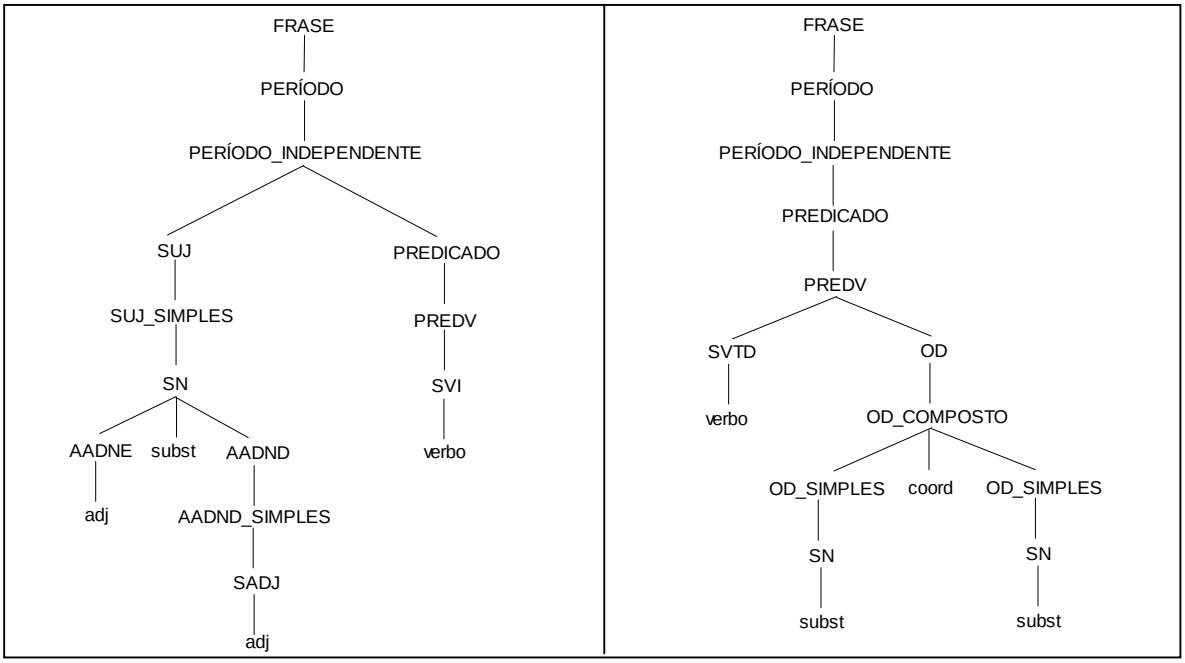

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTD
OD
AADNE
subst
verbo
OD_SIMPLES
SDET
SN
artigo
AADNE
subst
adj

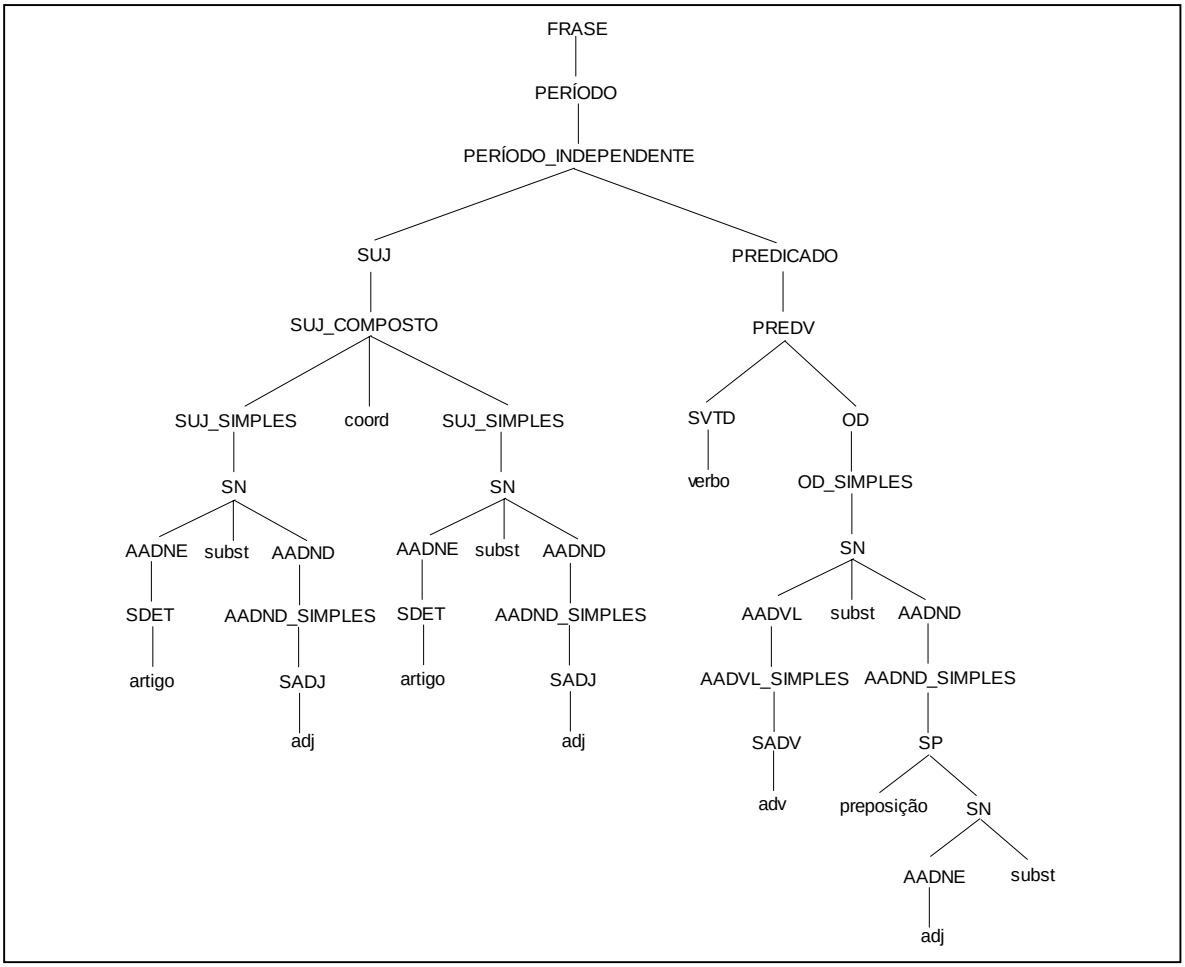

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDN
SN
SVL
PSUJ
pron_subst
verbo
PSUJ_SIMPLES
SADJ
AADVL
adj
CN
AADVL_SIMPLES
SP
SADV
preposição
SN
adv
AADNE
subst
AADND
adj
AADND_SIMPLES
SADJ
adj

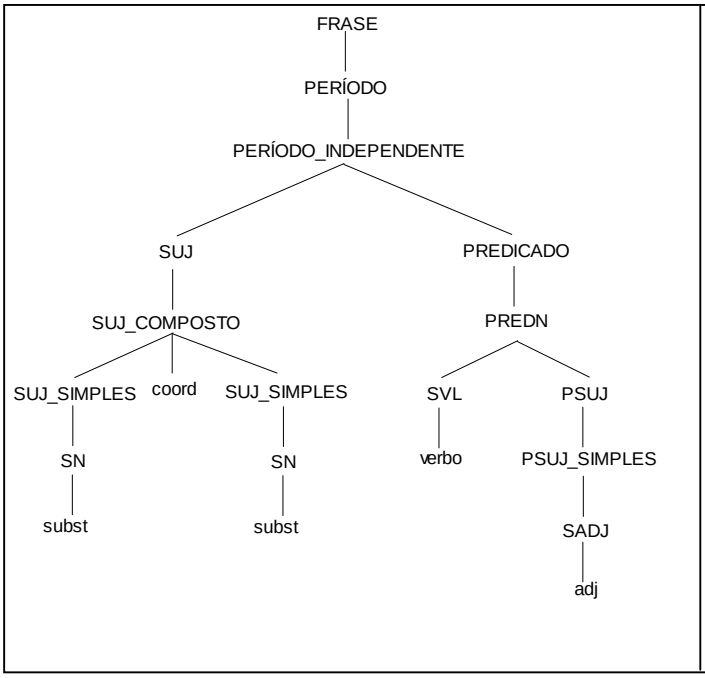

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDVN
SN
ODA
SVTD
PSUJ
pron_subst
lexico
verbo
PSUJ_SIMPLES
SADJ
AADVL
adj
AADVL_SIMPLES
SADV
adv

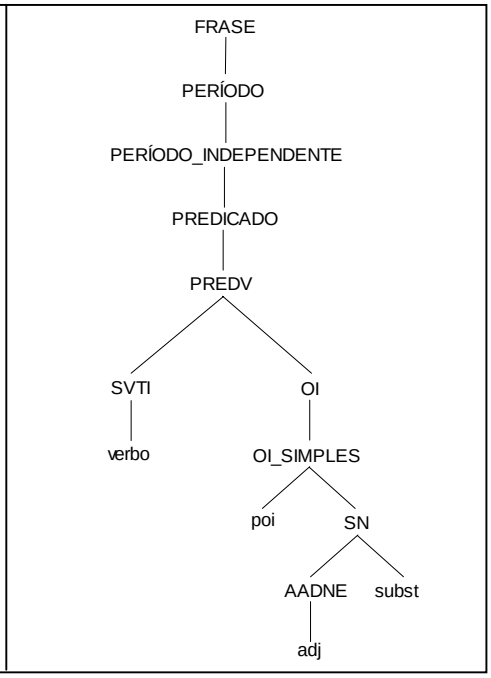

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTI
OI
AADNE
subst
verbo
OD_SIMPLES
SDET
poi
SN
artigo
subst
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTD
OD
pron-subst
verbo
OD_SIMPLES
SN
AADVL
subst
AADND
AADVL_SIMPLES
AADND_SIMPLES
SADV
SP
adv
preposição
SN
pron-subst

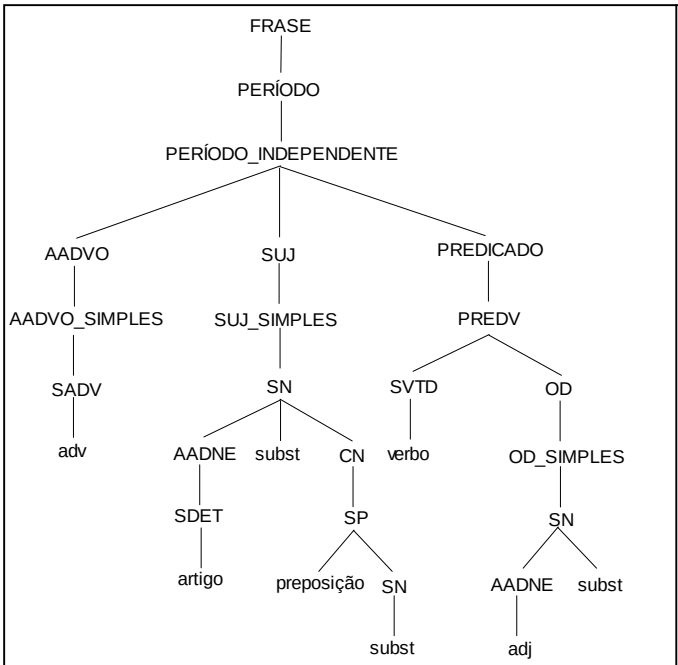

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVI
subst
AADND
`verboaux
verbo
AADND_SIMPLES
SADJ
adj
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTD
OD
AADNE
subst
CN
verbo
OD_SIMPLES
SDET
SP
SN
artigo
preposição
SN
AADNE
subst
subst
adj

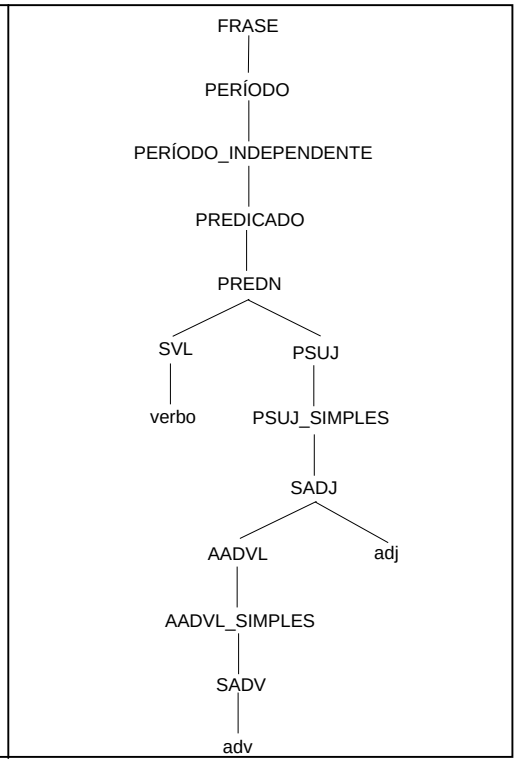

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVI
AADNE
subst
AADND
verbo
adj
AADND_SIMPLES
SADJ
adj
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTD
OD
verbo
OD_COMPOSTO
OD_SIMPLES
coord
OD_SIMPLES
SN
SN
subst
subst

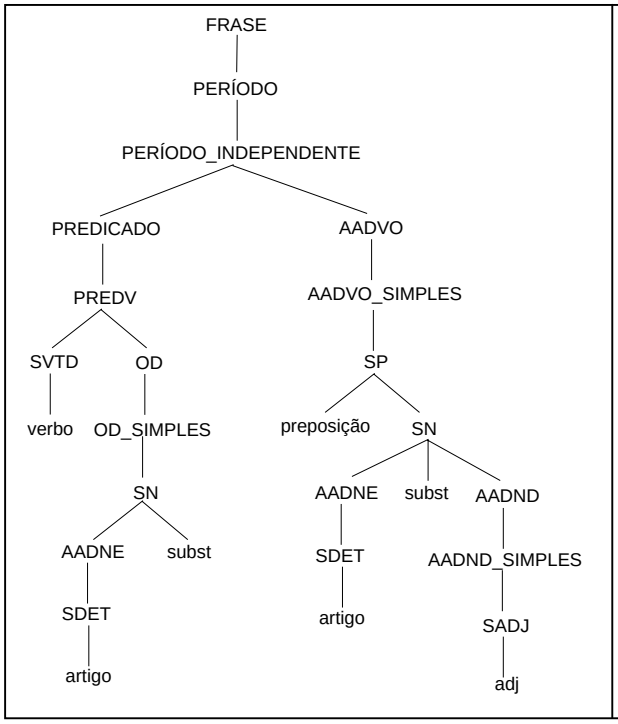

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_COMPOSTO
PREDV
SUJ_SIMPLES
coord
SUJ_SIMPLES
SVTD
OD
SN
SN
verbo
OD_SIMPLES
AADNE
subst
AADND
AADNE
subst
AADND
SN
SDET
AADND_SIMPLES
SDET
AADND_SIMPLES
AADVL
subst
AADND
artigo
SADJ
artigo
SADJ
AADVL_SIMPLES
AADND_SIMPLES
adj
adj
SADV
SP
adv
preposição
SN
AADNE
subst
adj

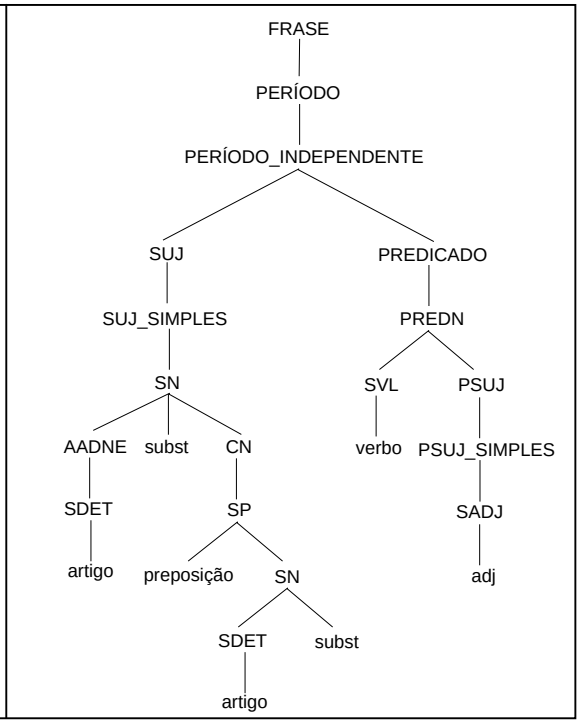

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_COMPOSTO
PREDN
SUJ_SIMPLES
coord
SUJ_SIMPLES
SVL
PSUJ
SN
SN
verbo
PSUJ_SIMPLES
subst
subst
SADJ
adj

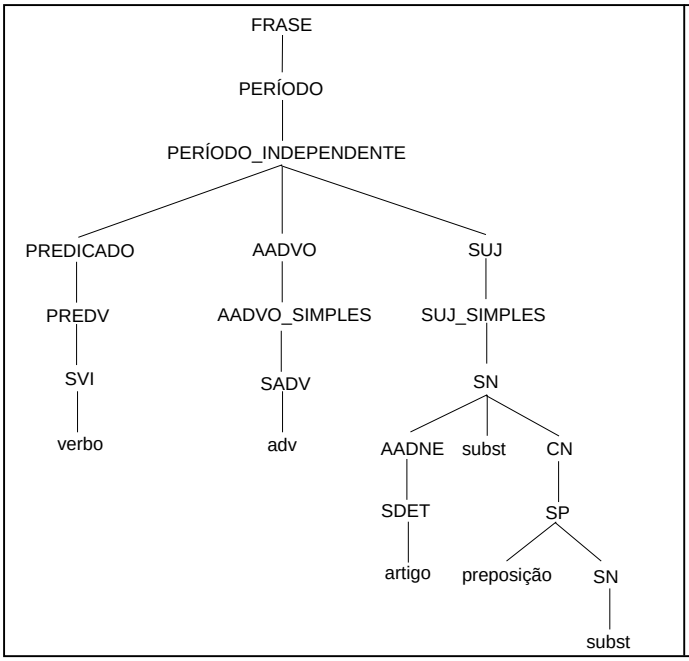

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTI
OI
verbo
OI_SIMPLES
poi
SN
AADNE
subst
adj

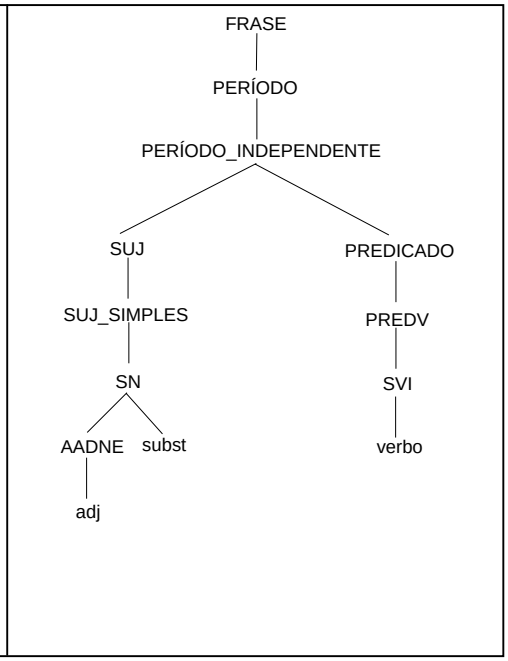

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
AADVO
SUJ
PREDICADO
AADVO_SIMPLES
SUJ_SIMPLES
PREDV
SADV
SN
SVTD
OD
adv
AADNE
subst
CN
verbo
OD_SIMPLES
SDET
SP
SN
artigo
preposição
SN
AADNE
subst
subst
adj

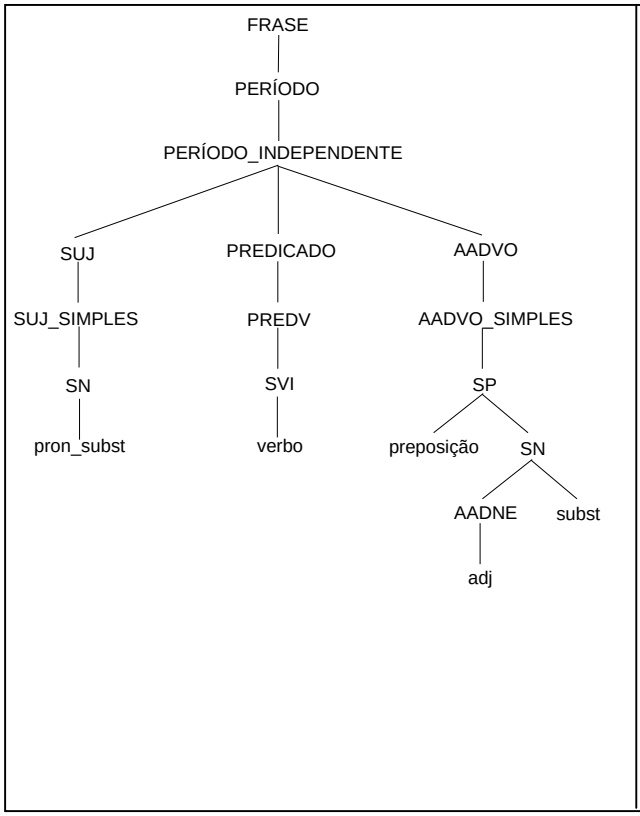

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDN
SVL
PSUJ
verbo
PSUJ_SIMPLES
SADJ
AADVL
adj
AADVL_SIMPLES
SADV
adv

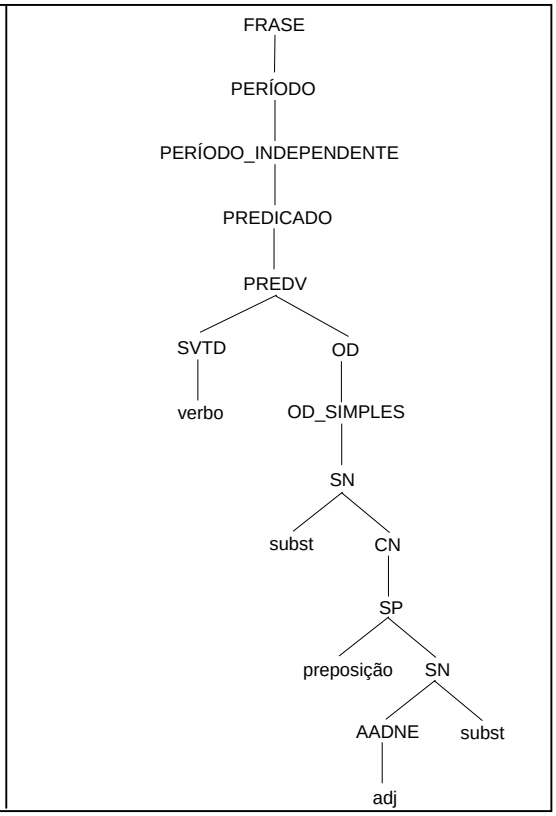

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
AADVO
PREDV
AADVO_SIMPLES
SVTD
OD
SP
verbo
OD_SIMPLES
preposição
SN
SN
AADNE
subst
AADNE
subst
AADND
AADNE
subst
SDET
SDET
AADND_SIMPLES
artigo
artigo
SADJ
adj

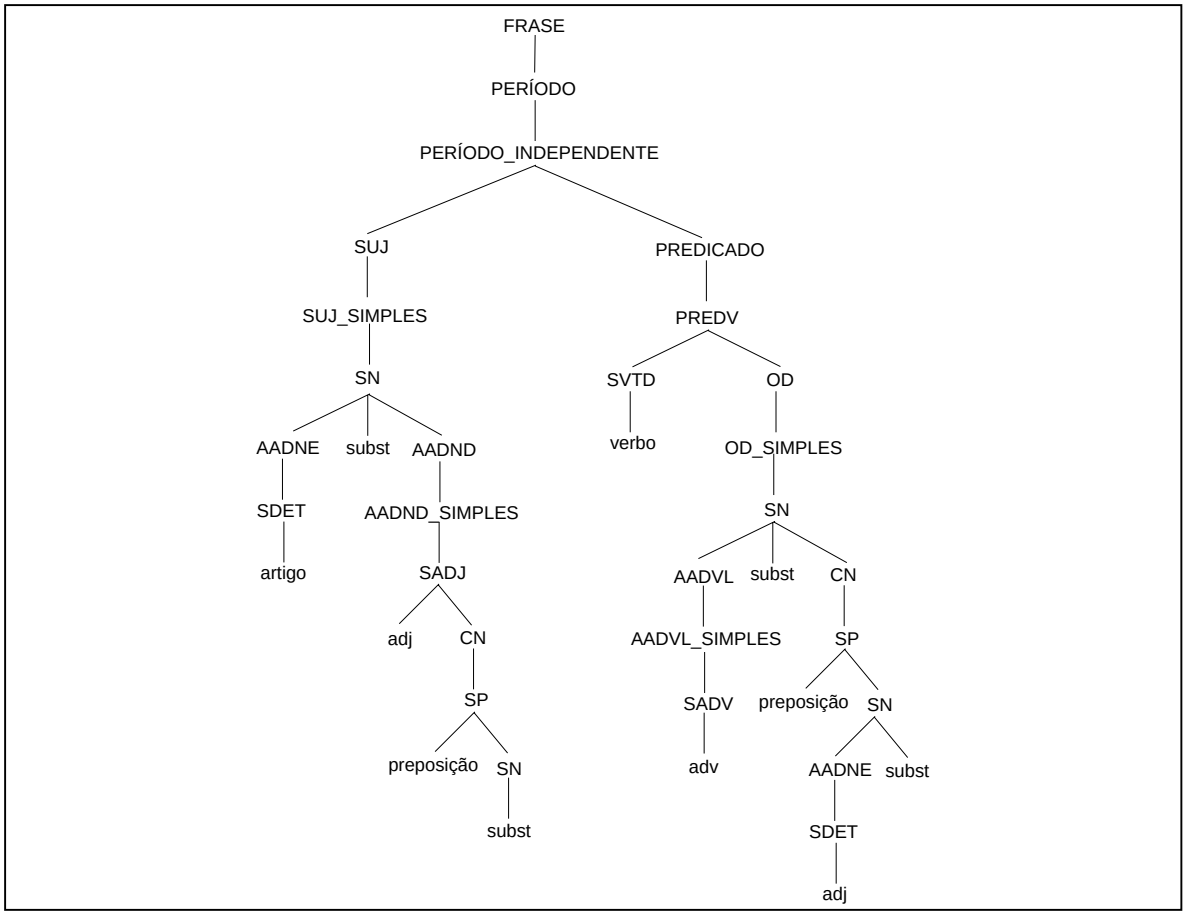

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDN
SN
SVL
PSUJ
AADNE
subst
CN
verbo
PSUJ_SIMPLES
SDET
SP
SADJ
artigo
preposição
SN
adj
SDET
subst
artigo

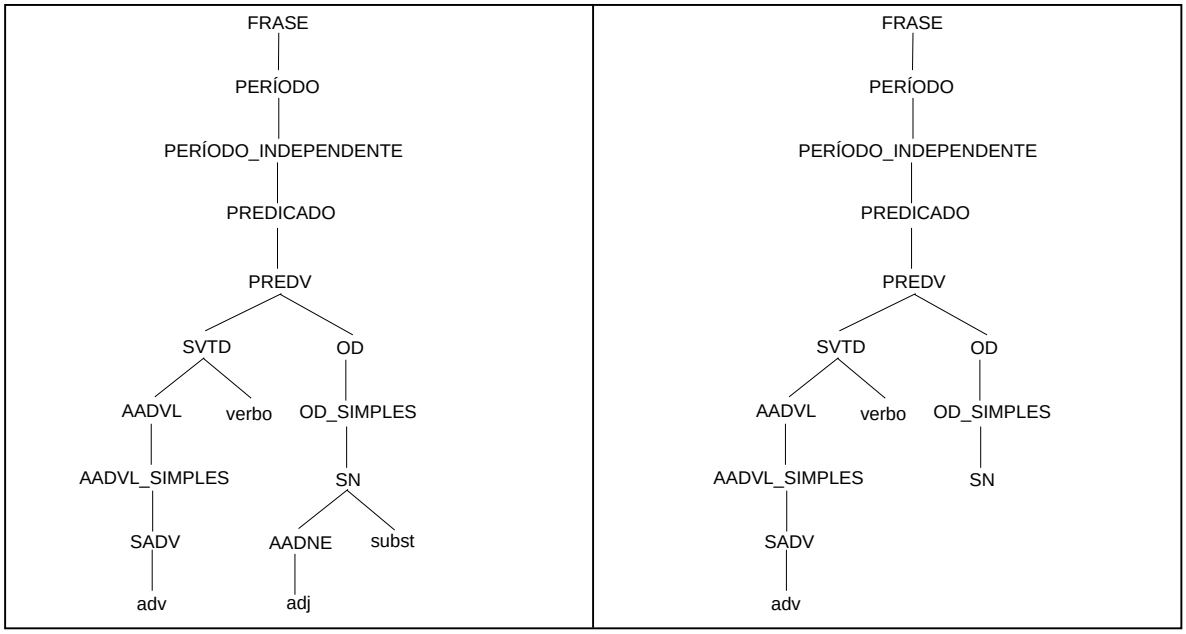

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
AADVO
SUJ
PREDV
AADVO_SIMPLES
SUJ_SIMPLES
SVI
SADV
SN
verbo
adv
AADNE
subst
CN
SDET
SP
artigo
preposição
SN
subst

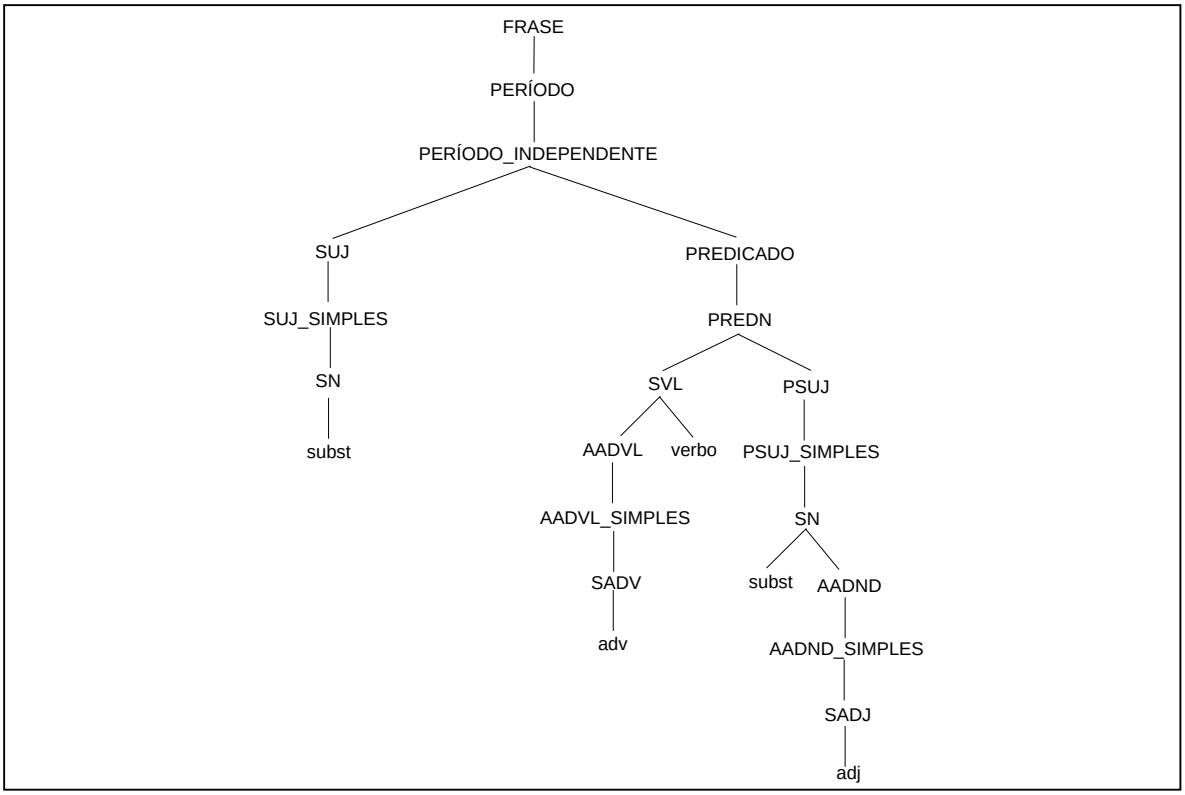

FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVI
AADNE
subst
verbo
adj


FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
AADVO
SUJ_SIMPLES
PREDV
AADVO_SIMPLES
SN
SVI
SP
pron_subst
verbo
preposição
SN
AADNE
subst
adj


FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTD
OD
verbo
OD_SIMPLES
SN
subst
CN
SP
preposição
SN
AADNE
subst
adj


FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDV
SN
SVTD
OD
AADNE
subst
AADND
verbo
OD_SIMPLES
SDET
AADND_SIMPLES
SN
artigo
SADJ
AADVL
subst
CN
adj
CN
AADVL_SIMPLES
SP
SP
SADV
preposição
SN
preposição
SN
adv
AADNE
subst
subst
SDET
adj


FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTD
OD
AADVL
verbo
OD_SIMPLES
AADVL_SIMPLES
SN
SADV
AADNE
subst
adv
adj
FRASE
PERÍODO
PERÍODO_INDEPENDENTE
PREDICADO
PREDV
SVTD
OD
AADVL
verbo
OD_SIMPLES
AADVL_SIMPLES
SN
SADV
adv


FRASE
PERÍODO
PERÍODO_INDEPENDENTE
SUJ
PREDICADO
SUJ_SIMPLES
PREDN
SN
SVL
PSUJ
subst
AADVL
verbo
PSUJ_SIMPLES
AADVL_SIMPLES
SN
SADV
subst
AADND
adv
AADND_SIMPLES
SADJ
adj