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

Descrição de um Protótipo de Realização Superficial do Português

Mauricio José Carvalho de Bem

Lucia Helena Machado Rino

NILC-TR-02-18

Setembro, 2002

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

Resumo

Apresentamos aqui a descrição completa de um realizador superficial baseado em decisões de estilo, construído pela prototipação de alguns casos particulares definidos a partir de sentenças do português. Etapas de projeto para a especificação do sistema já foram descritas em outros relatórios técnicos, motivo pelo qual contemplaremos aqui, sobretudo, a questão computacional.

Este trabalho conta com o apoio
financeiro do CNPq (PIBIC).

Índice

1. Introdução 1
2. Modelagem computacional 1
2.1. A geração dos dados de entrada do realizador 3
2.2. Construção de estruturas-f de sentenças de saída 7
3. Detalhes de implementação 12
4. Considerações finais 14
Referências Bibliográficas 15

Índice de Figuras

Figura 1 – Arquitetura do realizador superficial 2
Figura 2 - Árvore sintática de S1 3
Figura 3 - Exemplo de estrutura-c 4
Figura 4 - Estrutura-c com variáveis f associadas 5
Figura 5 - Descrição funcional de S1 5
Figura 6 - Estrutura-f final 6
Figura 7 - Estrutura-f correspondente a S1, na norma culta 8
Figura 8 - Estrutura sintática de S2 9
Figura 9 - Estrutura sintática de S2' 10
Figura 10 - Estrutura-f de S2 10
Figura 11 - Estrutura-f de S2' 11
Figura 12 - Exemplo de execução do programa 13
Figura 13 - Estrutura de uma estrutura-f linearizada 13

1. Introdução

Neste relatório iremos descrever a implementação do realizador superficial proposta como Projeto PIBIC/CNPq (de Bem, 2001), cujas etapas de especificação lingüística já foram descritas anteriormente (Espina et al., 2002; de Bem & Rino, 2002). Apresentaremos, primeiramente, a modelagem computacional (Seção 2), ilustrando, a seguir, como o sistema mapeia uma estrutura de entrada em estrutura de saída (Seção 3). Por fim, descrevemos algumas questões relevantes de implementação do protótipo (Seção 4).

2. Modelagem computacional

A modelagem do realizador superficial consiste em duas etapas básicas: a) especificar os recursos lingüísticos para que seja possível ao protótipo identificar problemas de estilo em dados de entrada do sistema e, a seguir, apontar possíveis soluções para os mesmos; b) especificar os processos lingüístico-computacionais correspondentes, para que, finalmente, sejam geradas as sentenças com mesmo potencial informativo, porém, com variações de estilo. Como já foi apontado em (de Bem e Rino, 2002), essas variações, no estágio atual do protótipo, são somente lexicais.

Os recursos envolvidos nessa etapa de modelagem são apontados na Figura 1. Detalharemos, particularmente, como foram especificadas as regras de estilo e de mapeamento de uma estrutura conceitual de entrada na estrutura conceitual da sentença que deverá ser realizada superficialmente, ilustrando o comportamento do protótipo com base em alguns exemplos.

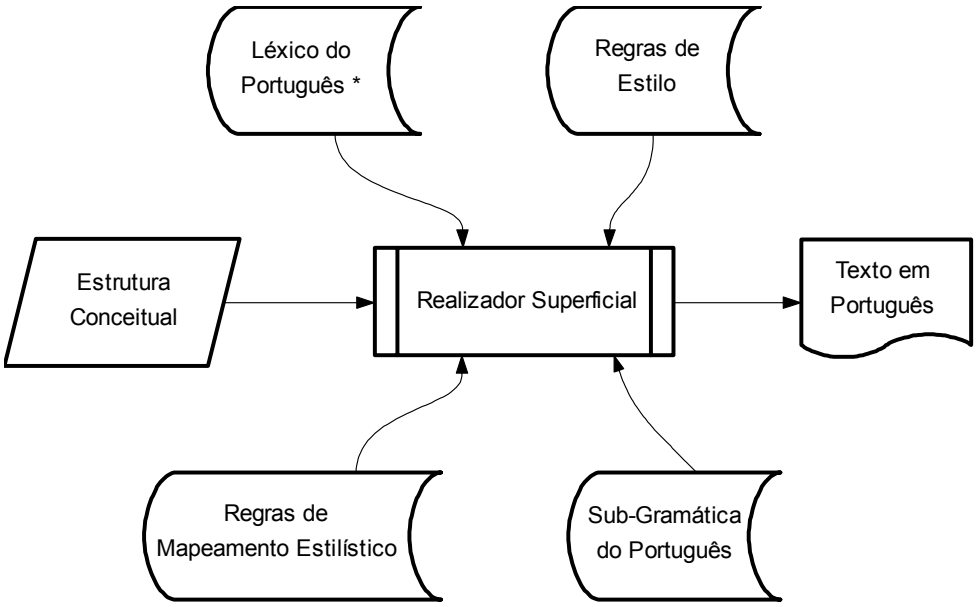

Figura 1 – Arquitetura do realizador superficial

Os dados de entrada do sistema, isto é, as estruturas conceituais, são estruturas funcionais, ou estruturas-f, representadas com base na *Lexical-Functional Grammar*, ou LFG (Kaplan & Bresnan, 1982). Essa mesma forma de representação será utilizada pelo realizador para gerar a sentença de saída.

Nesta seção, apresentaremos a forma como obtivemos, a partir de sentenças em português, suas estruturas-f, para, a seguir, descrever como se dá a unificação funcional das mesmas com as estruturas-f indicadas pelo sistema, para gerar as sentenças finais. Esse processo de unificação é descrito por Shieber (1986), em função da representação léxico-funcional, que é bastante conveniente quando adotamos um ambiente de programação que já incorpora tal processo, como veremos na Seção 3. A geração desses dados de entrada se baseou na escolha de um corpus, conforme descrito em (de Bem & Rino, 2002).

2.1. A geração dos dados de entrada do realizador

A geração da estrutura-f de cada sentença foi feita de forma manual, a partir de sua árvore sintática, produzida automaticamente pelo *parser* do português de Bick (2000)[1]. Essa etapa de análise não foi prevista, inicialmente, no trabalho de Iniciação Científica que envolve o projeto do realizador. Entretanto, buscamos otimizar as etapas de especificação de dados, visando, sobretudo, manter o foco na modelagem computacional para a realização superficial.

A partir da árvore sintática de uma sentença, obtivemos sua estrutura de constituintes, ou estrutura-c, que indica a configuração estrutural da sentença. Nesta etapa, foram usadas as regras gramaticais da sub-gramática em foco (já construída em função do corpus de sentenças adotado). Assim, para a sentença [S1] "Thunderbird inaugura a nova programação teen.", a árvore sintática gerada pelo *parser* é apresentada na Figura 2 e sua estrutura-c, na Figura 3. As regras gramaticais utilizadas para representar essa estrutura são apresentadas na Figura 4.

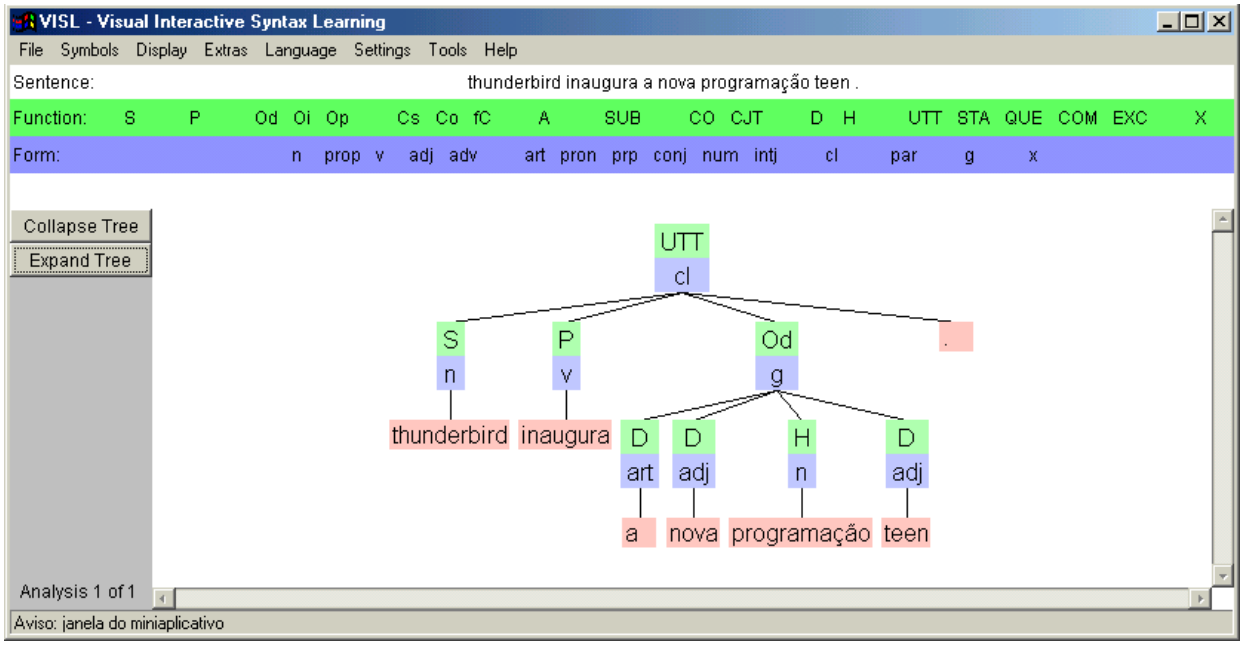

Figura 2 - Árvore sintática de S1

[1] http://visl.hum.sdu.dk/visl/pt/

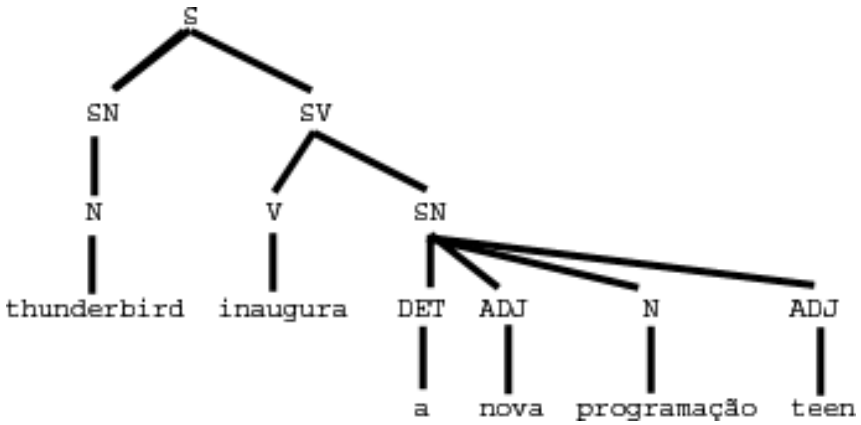

Figura 3 - Exemplo de estrutura-c[2]

```
s → sns(Gen,Num),sv(_).
sns(Gen,Num) → subs(Num,Gen,_,_).
sns(Gen,Num) → art(_,Num,Gen,_),adj(Num,Gen,_,_),subs(Num,Gen,_,_),
      adj(Num,Gen,_,_).
sv(Num) → v(Num,_,_,_),sns(Gen,_).
```

Figura 4 - Regras da sub-gramática

Após obtermos a estrutura-c, associamos variáveis (f_i) aos seus símbolos não terminais (Figura 5), para gerar, então, sua descrição funcional, como mostra a Figura 6.

[2] ADJ:adjetivo; DET:artigo; N:substantivo; S:sentença; SN:sintagma nominal; SV:Sintagma Verbal; V:verbo.

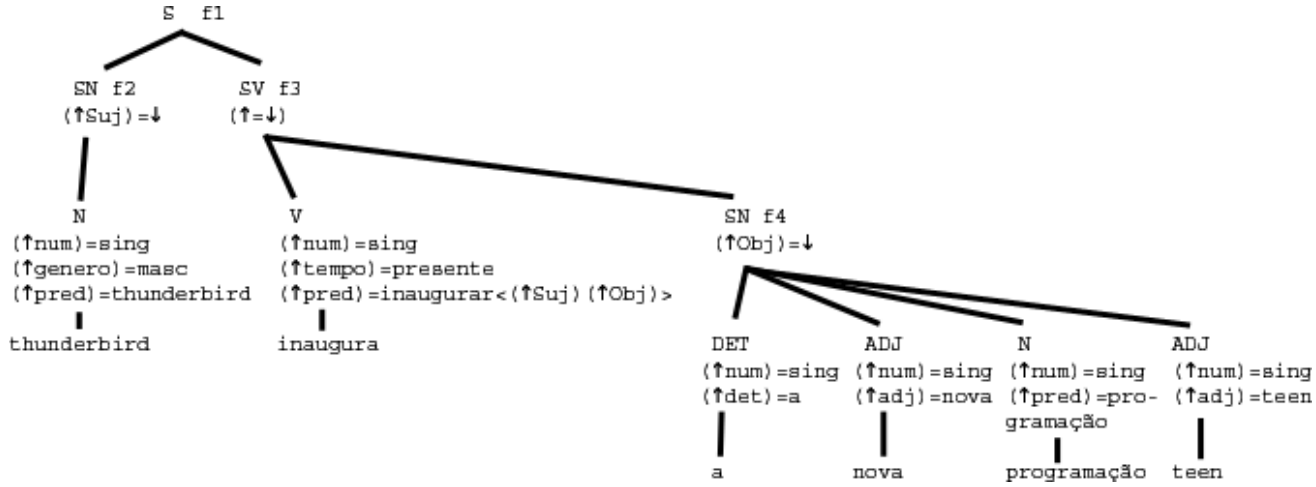

Figura 4 - Estrutura-c com variáveis f associadas

```
(f1Suj)=f2
f1=f3
(f2num)=sing
(f2genero)=masc
(f2pred)='thunderbird'
(f3num)=sing
(f3tempo)=presente
(f3pred)='inaugurar<(↑Suj)(↑Obj)>'
(f3Obj)=f4
(f4num)=sing
(f4det)='a'
(f4adj)={'nova','teen'}
(f4genero)=masc
(f4pred)='programação'
```

Figura 5 - Descrição funcional de S1

Essa descrição funcional é expressa graficamente na forma apresentada na Figura 7. É essa estrutura que serve de entrada para o realizador superficial.

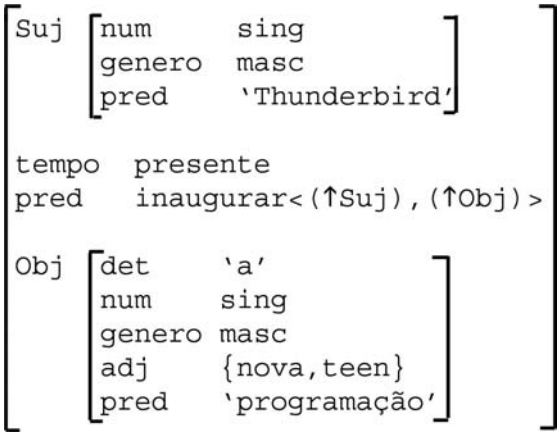

Figura 6 - Estrutura-f final

Com esse exemplo, podemos notar que temos conjuntos funcionais distintos para cada grande componente sintático[3], o qual é identificado pelo seu nome de *slot*, com valor correspondente. Pode haver, ainda, mais de um nível de nomes de *slots*. Por exemplo, para a representação do sujeito (Suj), temos os campos *num, gênero* e *pred*, que indicam, respectivamente, os traços morfológicos de número e gênero do objeto, assim como seu valor (predicação). Seus *slots* são preenchidos com os valores obtidos a partir da análise sintática da sentença em português. Além de traços morfossintáticos, há, assim, a associação direta com o item lexical dicionarizado. Esses valores são indicados entre apóstrofes, como nos campos *pred* e *det* da Figura 7.

Estruturas-f como a ilustrada são centrais em nosso realizador superficial, devido ao seu alto grau de computabilidade e à facilidade de representação da estrutura sintática de sentenças da língua natural. Além disso, elas permitem a associação direta com o léxico, incorporando, inclusive, estruturas de relativa complexidade. Veremos, na próxima seção, que outra vantagem de adotarmos estruturas-f como forma de representação conceitual dos dados de entrada do realizador está no seu processamento, mais especificamente, na possibilidade de usar diretamente o processo de unificação para obter a estrutura de saída.

[3] Componentes sintáticos são iniciados por letras maiúsculas na Figura 7.

2.2. Construção de estruturas-f de sentenças de saída

Como vimos anteriormente, a estrutura-f incorpora as relações gramaticais, funcionais, entre os componentes de uma sentença.

O aspecto funcional dessa estrutura é importante na realização superficial devido ao fato dela ser de fácil manipulação automática, o que possibilita fazermos uma análise profunda e conseqüentemente podermos encontrar o desvio estilístico como descrito a seguir.

O mapeamento estilístico entre uma estrutura-f de entrada e uma estrutura-f de saída é realizado da seguinte maneira: primeiramente, determina-se o problema de estilo da estrutura-f de entrada, para isso, o sistema recorre às definições de possíveis figuras de estilo do português, expressas no recurso lingüístico 'Regras de Estilo'. Uma vez identificado o problema, buscam-se possíveis variações que legitimem as decisões para a produção da sentença de saída. Esse processo de mapeamento se baseia nas regras definidas em outro repositório lingüístico: 'Regras de Mapeamento Estilístico'. É escolhida, então, uma das regras de mapeamento, cuja execução resulta na escolha dos itens lexicais que substituirão os originais, isto é, aqueles com desvio estilístico. A seguir, esses dados resultantes do processo de mapeamento são unificados, segundo o método proposto por (Elhadad, 1991), em um template, que será a estrutura-f de saída, juntamente com os dados da estrutura-f de entrada.

Para a sentença S1, por exemplo, o realizador superficial identifica como desvio da norma de estilo do português o uso de estrangeirismo[4] – item lexical *teen* (vide Figura 7). Este item é substituído, a partir de uma regra possível de mapeamento já definida no sistema, pelo item lexical *jovem,* como ilustra a Figura 8.

[4] Para detalhes sobre figuras de estilo, vide (Espina et al., 2002).

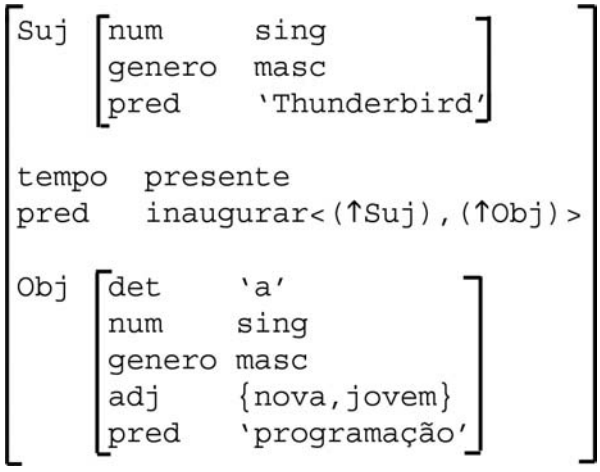

Figura 7 - Estrutura-f correspondente a S1, na norma culta

Neste exemplo, a alteração é simplesmente lexical e, assim, a estrutura-f resultante não sofre nenhum tipo de alteração sintática. Somente o valor do campo obj(adj) é alterado para {*nova*, *jovem*}, ou seja, há duas possibilidades de realização lingüística, trocando-se o *teen* por outro item lexical. Como podemos observar, o campo pred (*programação*) do objeto (Obj) da sentença S1 passa a ter, assim, a modificação por dois adjetivos licenciados na norma culta: o mesmo já empregado na entrada (*nova*) e o que substitui o estrangeirismo empregado antes (*jovem* para *teen*).

Um outro exemplo do mapeamento estilístico efetuado pelo realizador superficial é dado para a sentença [S2] "Cresceram às custas de São Paulo", a qual apresenta um problema de uso da locução prepositiva "*às custas de*", a qual deve ser substituída por "*à custa de*", considerando-se sua variação para a norma culta, com alto grau de permissividade. Sendo [S2'] a sentença modificada "Cresceram *à custa de* São Paulo", ambas as árvores sintáticas (geradas automaticamente pelo *parser*, no estágio pré-processamento) reproduzidas abaixo, nas Figuras 9 e 10, mostram que será necessária uma transformação estrutural, para a geração de S2' a partir de S2. Neste caso, a estrutura-f de entrada do protótipo (Figura 11), deverá ser modificada para a estrutura-f de S2' (Figura 12). Podemos ver que a modificação estrutural

exigida, neste caso, recai sobre o campo ‘prep’: na estrutura-f de S2, este é associado ao campo det (valor do slot ‘as’) e tem valor de slot também ‘a’ (preposição às = a + as); na estrutura-f de S2’, esse campo deve ser preenchido com o item lexical que indica a norma culta, para ‘*à custa de’*. Em nosso léxico, esta expressão é representada como uma locução prepositiva e, assim, ela será inteiramente usada como valor do *slot* correspondente do campo ‘prep’, como indica a Figura 12. De forma correspondente, serão também atualizados os campos de número e gênero do grande slot que representa o complemento sentencial (i.e., Compl).

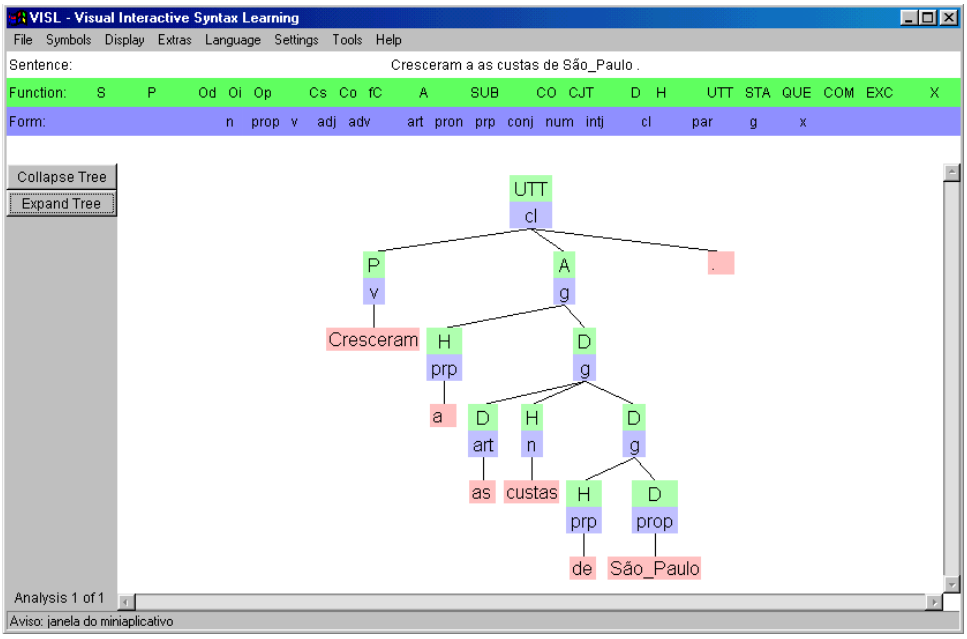

Figura 8 - Estrutura sintática de S2

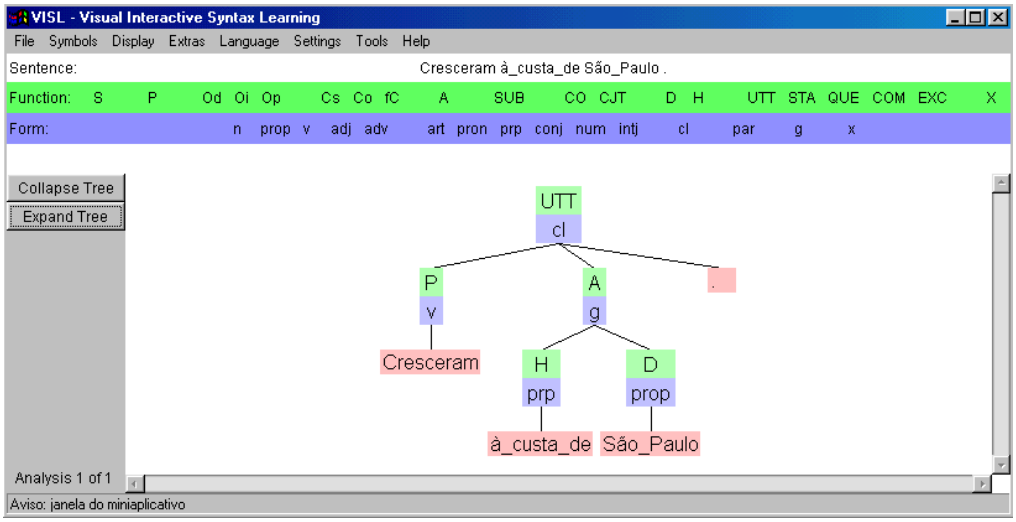

Figura 9 - Estrutura sintática de S2'

```
[ Suj [num  plural]
      [pred e      ]

tempo  pretérito perfeito
pred   crescer<(↑Suj),(↑Compl)>

Compl [prep  'a'                          ]
      [                                   ]
      [NP  [num    plural                 ]
           [genero fem                    ]
           [det    'as'                   ]
           [pred   'custas'               ]
           [                              ]
           [PP  [num    sing              ]
                [genero masc              ]
                [prep   'de'              ]
                [pred   'São Paulo'       ]]]]
```

Figura 10 - Estrutura-f de S2

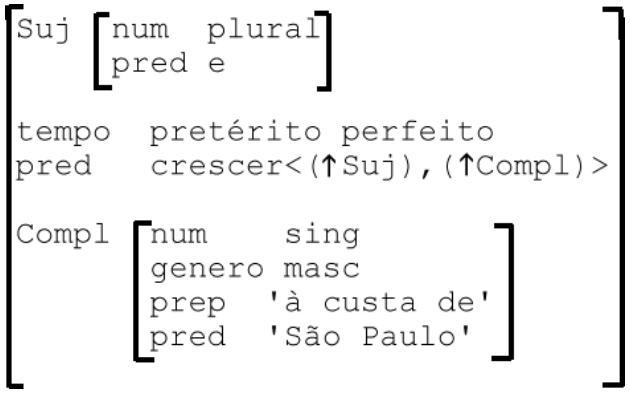

Figura 11 - Estrutura-f de S2'

Neste caso, portanto, a alteração não é simplesmente lexical, como no caso anterior e, por isso, outro conjunto de regras de mapeamento estilístico será acionado no protótipo, para a geração de S2', agora visando especificamente a alteração estrutural, também garantindo a produção de sentenças licenciadas pela norma da língua portuguesa.

Para que o realizador possa tomar decisões como as ilustradas aqui, o suposto usuário deve fornecer alguns dados adicionais, que lhe permitam recuperar o contexto de transformação possível da estrutura-f de entrada para uma estrutura-f licenciada pelas regras de mapeamento que o compõem, para, somente então, ser capaz de gerar a sentença correspondente, expressando a variação estilística escolhida. Detalhes sobre as formas de decisão que o sistema toma são dados a seguir.

3. Detalhes de implementação

Para a implementação do realizador escolhemos a linguagem Prolog (Clocksin & Mellish, 1984; Marcus, 1986), devido ao seu potencial de representação do conhecimento e à sua incorporação do Princípio de Resolução, o qual permite a unificação automática das estruturas-f em foco no sistema.

Entradas para o realizador devem ser expressas em arquivo, na forma linearizada (como uma lista em Prolog). O usuário pode escolher uma estrutura-f por vez para processar, assim como o tipo de mapeamento a ser realizado.

Para a sentença S1 ilustrada anteriormente – "Thunderbird inaugura a nova programação teen." – o usuário pode escolher uma das cinco opções fornecidas automaticamente, como mostra a Figura 13. Nesse exemplo, ao escolher a opção 1, o protótipo fará a realização superficial para a norma culta mas com um "comportamento" diferenciado em relação à opção 2, que também fará a realização para a norma culta, mas para os padrões do estado de São Paulo.

A opção 3 fará a realização lingüística para um nível mais ou menos culto da língua. A opção 4 permite ao usuário carregar uma estrutura-f de um arquivo em memória, para a realização lingüística. Um exemplo de estrutura-f presente em um arquivo de entrada é dado na Figura 14. Finalmente, a opção 5 permite ao usuário terminar a aplicação de realização lingüística.

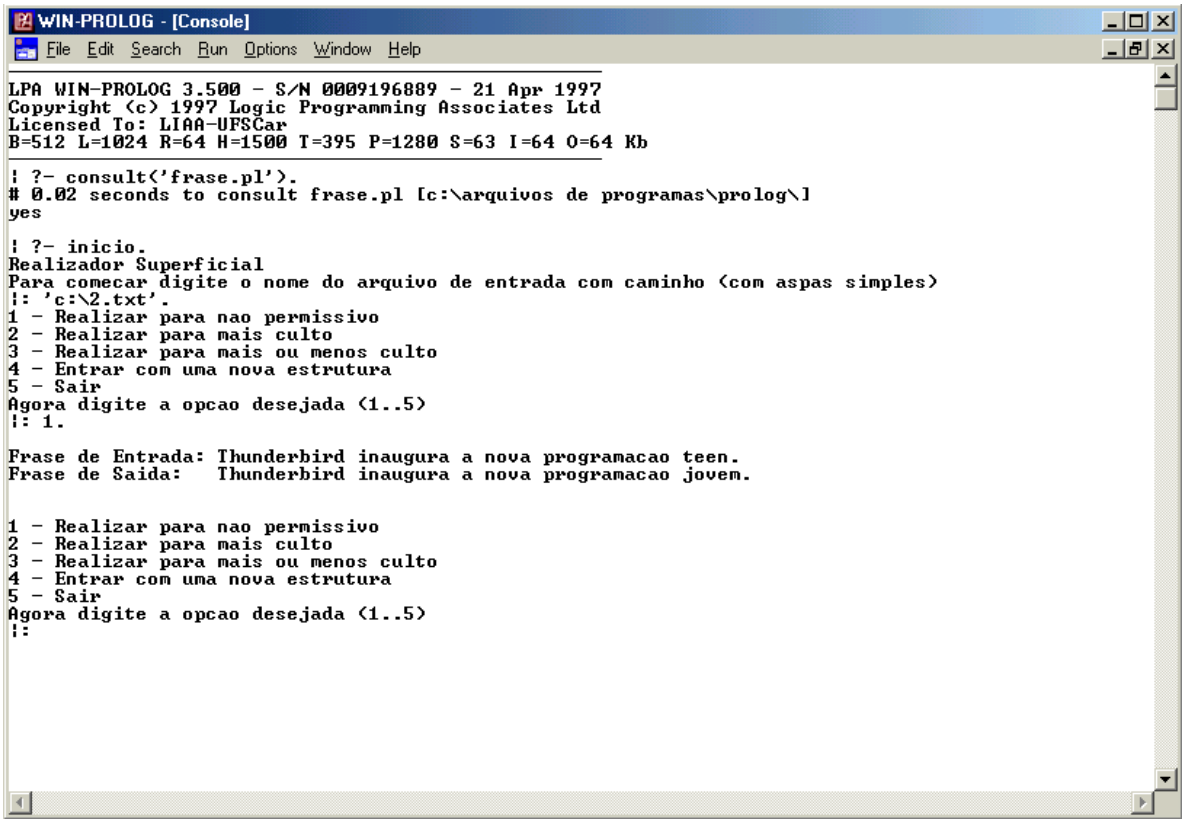

Figura 12 - Exemplo de execução do programa

```
[suj,[num(sing),genero(masc),pred(thunderbird)],num(sing),pessoa(terc),tempo(pres)
,pred(inaugurar),obj,[num(sing),genero(fem),det(a),adj(nova),adj(teen),pred(program
acao)]]
```

Figura 13 - Estrutura de uma estrutura-f linearizada

4. Considerações finais

Como já mencionamos, o protótipo ilustrado neste relatório somente considera variações de estilo que não sofrem modificações estruturais. Estas não foram implementadas devido à sua complexidade, especialmente em relação à necessidade de se expandir a sub-gramática do português, a qual deve passar a prever não só variações lexicais, mas também variações sintáticas. Neste caso, o recurso de regras de mapeamento estilístico deverá ser aumentado, para relacionar estruturas-f de entrada e estruturas-f de saída. Tais variações levarão, invariavelmente, a novas escolhas lexicais, implicando um aumento do número de entradas do próprio léxico.

Mesmo considerando somente o corpus originalmente selecionado, apontado em (de Bem & Rino, 2002) com sendo aquele sem simplificação de qualquer natureza, já podemos traçar algumas características de sintaxe que sofreriam modificações em nosso protótipo. Por exemplo, na sentença [S2] "Cresceram às custa de São Paulo.", a figura de estilo indicada é um problema no uso de locuções prepositivas). Se o tipo de realização escolhido pelo usuário fosse, p.ex., a realização mais culta, essa sentença sofreria uma alteração de sintaxe que seria a troca dos itens lexicais "*às custas de*" , por um item lexical que é "*à custa de*".

Uma importante observação a respeito da implementação do protótipo é que somente as sentenças que não sofrem modificações estruturais foram contempladas, ou seja, somente as sentenças que têm como saída a mesma estrutura-f da sentença de entrada foram implementadas. Entretanto, em um estágio mais avançado, o protótipo deverá prever, também, variações estruturais.

Referências Bibliográficas

de Bem, M. J. C. (2001). *Um protótipo de realizador superficial para decisões de estio no português*. Projeto proposto para bolsa PIBIC/CNPq. DC/UFSCar (Agosto/2001 – Julho/2002).

de Bem, M. J. C., Rino, L. H. M. (2002). *Especificação Lingüística de um Realizador Superficial Baseado em Decisões de Estilo*. Série de Relatórios Técnicos do NILC, NILC-TR-02-17. Setembro.

Bick, E. (2000). *The Parsing System Palavras: automatic grammatical analysis of Portuguese in a constraint grammar framework*. Arhus University Press. Tese de Doutorado.

Clocksin, W.F., Mellish, C.S. (1984). *Programming in Prolog*. Second Edition. Springer-Verlag, Germany.

Elhadad, M. (1991). *FUF: The Universal Unifier User Manual*. Version 5.0. Department of Computer Science, Columbia University. New York.

Espina, A.P.; de Bem, M.J.C.; Rino, L.H.M. (2002). *A exploração de questões de estilo do português para a realização superficial automática*. Série de Relatórios Técnicos do NILC, NILC-TR-02-16. Setembro.

Kaplan, R., Bresnan, J. (1982). *Lexical-Functional Grammar: a Formal System for Grammatical Representation*. In Bresnan, J. (ed.), The Mental Representation of Grammatical Relations, pp. 173-281. MIT Press, Cambridge MA.

Marcus, C. (1986). *Prolog Programming*. Arity Corporation. Addison-Wesley Publishing Company.

Shieber, S.M. (1986). *An Introduction to Unification-Based Approaches to Grammar*. *CSLI Lecture Notes* 4. University of Chicago Press.