**Instituto de Ciências Matemáticas e de Computação**

ISSN - 0103-2569

# Documentação e Atualização do Regra

**Thiago Alexandre Salgueiro Pardo**
**Lúcia Helena Machado Rino**
**Maria das Graças Volpe Nunes**

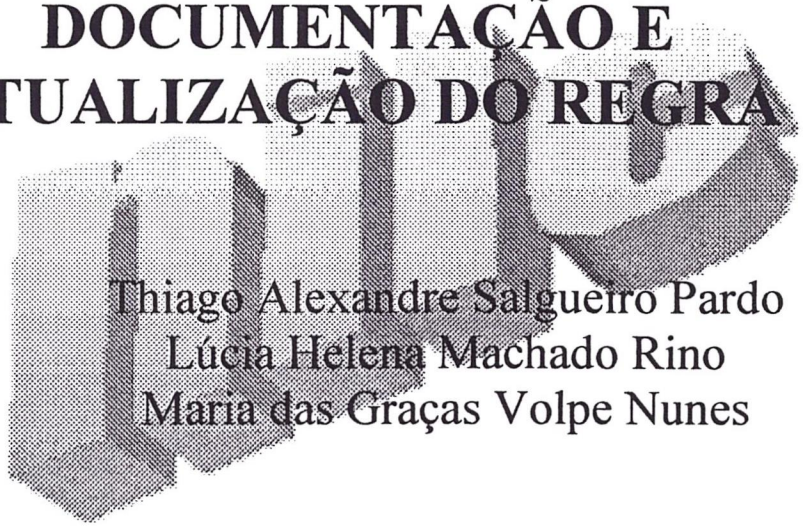
**Nº 109**

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Mar./2000

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

# DOCUMENTAÇÃO E ATUALIZAÇÃO DO REGRA



Thiago Alexandre Salgueiro Pardo
Lúcia Helena Machado Rino
Maria das Graças Volpe Nunes

**NILC-TR-00-3**

Março, 2000

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

# DOCUMENTAÇÃO E ATUALIZAÇÃO DO REGRA*

**Resumo**

É apresentado aqui o trabalho de documentação e atualização do Revisor Gramatical ReGra, desenvolvido pelo NILC e comercializado pela Itautec-Philco S.A. Após 5 anos de desenvolvimento, e a produção de várias versões do Revisor, sua documentação se tornou precária e insuficiente. Pela complexidade e tamanho do programa, essa tarefa se tornou, por si só, um processo complexo, para o qual foi necessária a aplicação de técnicas adequadas de engenharia reversa e sua documentação.

## 1. Introdução

O REvisor GRAmatical ReGra, comercializado desde 1995 pela Itautec/Philco, é resultado de um projeto de parceria universidade-empresa cujo desenvolvimento se dá há aproximadamente oito anos pela equipe do NILC/São Carlos – Núcleo Interinstitucional de Lingüística Computacional. No seu desenvolvimento, foi utilizada uma estratégia fundamentalmente dirigida pela aplicação: uma abordagem pontual permite a localização e a correção rápida de erros padronizados, enquanto que uma abordagem lingüística, voltada para a identificação das funções sintáticas e das relações de dependência de cada uma das palavras da sentença, permite a detecção e sugestão de correção de erros originários de fenômenos gramaticais. Entretanto, a qualidade da análise sintática e da correção gramatical é prejudicada pela ambigüidade categorial e pela ausência de informações adicionais para as palavras do léxico. Em especial, o fenômeno sintático em foco para tentar aprimorar tais versões consistiu na desambigüização lexical, que, de cunho semântico, deveria ser detectada durante o processamento sintático para se poder decidir criteriosamente sobre a classe gramatical das palavras.

Passando por várias versões até chegar à atual, o ReGra mantém as características originais de seu projeto no que diz respeito ao processamento morfológico e à correção ortográfica, mas apresenta várias melhorias em relação à análise sintática (*parsing*), dentre as quais destacam-se a reformulação do *parser* e a implementação de novas regras de correção gramatical e de desambigüização lexical, para a determinação das categorias gramaticais corretas das palavras. Essas atividades tiveram por objetivo principal o aprimoramento do ReGra através de (a) a redução do número de omissões da ferramenta quando a correção é necessária - falsos acertos; (b) redução do número de intervenções desnecessárias - falsos erros - e (c) atualização progressiva das possíveis regras de revisão gramatical para admitir a incorporação crescente das construções gramaticais da língua portuguesa.

O trabalho aqui relatado teve por objetivo fornecer uma documentação detalhada do código do ReGra, em sua versão atual, para permitir maior domínio da ferramenta, pelo acesso às suas características e falhas. Desse modo, a documentação visou possibilitar diagnósticos do comportamento inadequado do ReGra, assim como seu futuro aprimoramento, tanto em relação à revisão gramatical, quanto em relação ao processamento computacional, propriamente dito. Além da elaboração da documentação, este trabalho explorou alguns métodos para o aprimoramento do ReGra,

* Este trabalho foi desenvolvido no âmbito do projeto PADCT-III/Finep/Itautec-Philco S.A., # 03-CE-01/98-01/02-10.

avaliando a manipulação de informação semântica e o uso de métodos estatísticos para a otimização dos processos de *parsing* e *checking*.

A Seção 2 apresenta o modo como a documentação foi elaborada e a Seção 3 discute o trabalho que está sendo realizado para o aprimoramento do ReGra. As conclusões e perspectivas de continuidade deste trabalho são apresentadas na Seção 4.

## 2. Documentação do ReGra

O ReGra foi documentado segundo o modelo Fusion/RE (Penteado, 1996), de modo que a documentação é formada, principalmente, pela revitalização da arquitetura do sistema e por seu modelo operacional.

### 2.1. Arquitetura do ReGra

A arquitetura (ou organização funcional) do ReGra pode ser esquematizada em módulos de funcionalidade e nas relações entre eles, conforme ilustra a Figura 1.

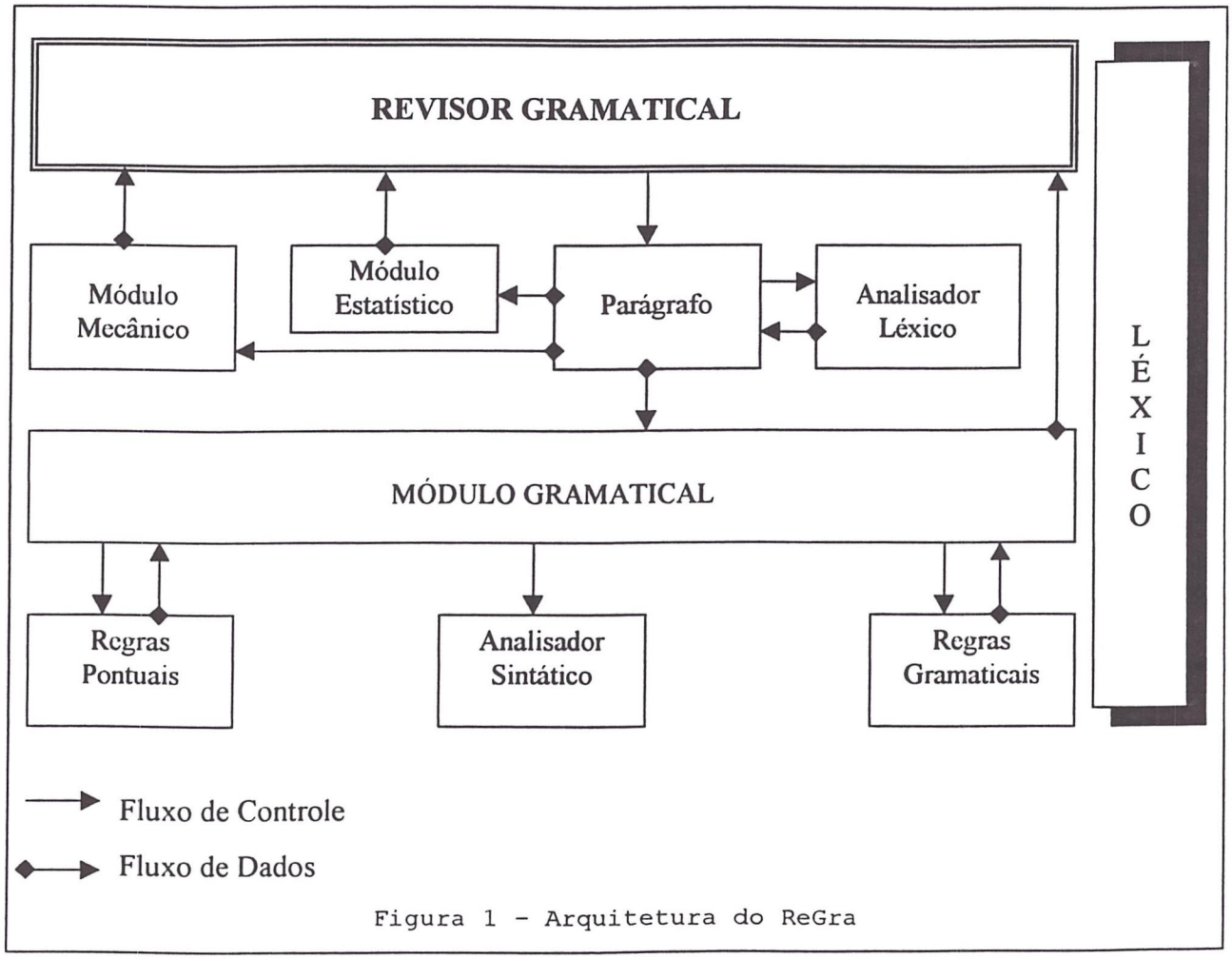



Figura 1 - Arquitetura do ReGra

Desta forma, quatro tipos de processamento caracterizam o funcionamento do ReGra, a saber:

- Lexical: composto pelas funções de acesso e recuperação de informação lexical, sendo estas informações utilizadas durante todo o processamento para a revisão gramatical.
- Estatístico: responsável por coletar medidas estatísticas de um texto e calcular o índice de legibilidade do mesmo, indicando o seu nível de dificuldade de leitura.

- Mecânico: identifica e trata alguns tipos de erros facilmente identificáveis que normalmente não são detectados por um corretor ortográfico, tais como o balanceamento de delimitadores (parênteses, chaves, colchetes e aspas) e o espaçamento entre componentes sentenciais, além da pontuação irregular em uma sentença.
- Lingüístico: subdividido em três partes:
  - Regras pontuais: responsáveis por corrigir erros pré-determinados quando estes são encontrados em um parágrafo;
  - Análise sintática (*parsing*): realiza a análise sintática do texto sob análise, parágrafo a parágrafo;
  - Verificação gramatical (*checking*): responsável pela identificação das relações de dependência entre as palavras de um parágrafo e, caso necessário, pela aplicação de regras de correção adequadas ao erro encontrado.

## 2.2. Modelo de Operações

Segundo o método Fusion/RE, o modelo de operações é o conjunto de formulários que possuem informações detalhadas sobre os componentes do sistema sob análise. As figuras 2 e 3 mostram, respectivamente, um exemplo de modelo de operações para um módulo e uma rotina do ReGra.

*Nome do módulo:* adj_adv.h
*Descrição:* Módulo que declara as classes adj_adv, aadv_simples e aadv_composto e seus métodos.

```
class adj_adv : public lexico
{
protected:
    aadv_simples *adjadv_simples;
    aadv_composto *adjadv_composto;
public:
    adj_adv(paragrafo *);
    ~adj_adv();
    int regra(int);
    int nuc_adj_dir(data_erro *data);
    int nuc_adj_esq(data_erro *data);
    int tem_paadv();
};
class aadv_simples  : public basica
{
protected:
    nuc_suj *nucsuj;
    subordinada *ossadv;
    aposto *Aposto;
    nucleo *nuc;
public:
    aadv_simples(paragrafo *);
    ~aadv_simples();
    int regra(int);
    int nuc_adj_dir(data_erro *);
    int nuc_adj_esq(data_erro *);
    int tem_paadv();
};
class aadv_composto : public  basica
{
protected:
    conjunc *cc;
    nuc_suj *nucsuj1, *nucsuj2, *nucsuj3;
public:
    aadv_simples *aadv1;
    aadv_simples *aadv2;
    aadv_composto *aadv_comp;
    aadv_composto(paragrafo *);
    ~aadv_composto();
    int regra(int);
    int nuc_adj_dir(data_erro *);
    int nuc_adj_esq(data_erro *);
    int tem_paadv();
};
```

Figura 2 - Descrição do módulo adj_adv.h

*Procedimento:* aadv_composto::regra

*Módulo:* adj_adv.cpp

*Descrição:* Método que verifica a ocorrência de adjunto adverbial composto.

*Entradas:*
int pos

*Saídas:*
int

*Chama:*
lexico::ignora_delimitadores_sintagma
lexico::verifica_adjetivo
lexico::verifica_adj_genero
lexico::verifica_adj_numero
conjunc::tipo
lexico::verifica_adverbio
aadv_composto::paadv
nuc_suj::regra3
nuc_suj::regra4
nuc_suj::regra2
nuc_suj::regra5
aadv_simples::regra
aadv_simples::tem_paadv
aadv_composto::regra

*Chamado por:*
adj_adv::regra
aadv_composto::regra

*Sobrecarga:* Não

Figura 3 - Descrição da Rotina regra da Classe aadv_composto

## 2.3. Estruturação das Regras Gramaticais do ReGra

As regras sintáticas do ReGra são modeladas como ATNs (*Augmented Transition Networks*) (Woods, 1970). A Figura 4 exibe uma ATN para o *parsing* de um adjunto adverbial com a seguinte formação:

```
preposição + [adjunto adnominal a esquerda] + núcleo + aposto
+ [adjunto adnominal a direita]

                              ou

preposição + [adjunto adnominal a esquerda] + núcleo +
[adjunto adnominal a direita]
```

Nessa notação, os delimitadores "[" e "]" indicam termos opcionais, os quais são mapeados para arcos sem rótulos na ATN. O círculo apontado por uma seta sem origem e o círculo com a borda mais grossa indicam, respectivamente, os pontos inicial e final da ATN, ou seja, o início e o fim da especificação de *parsing* do constituinte gramatical indicado pela ATN, no caso, o adjunto adverbial.

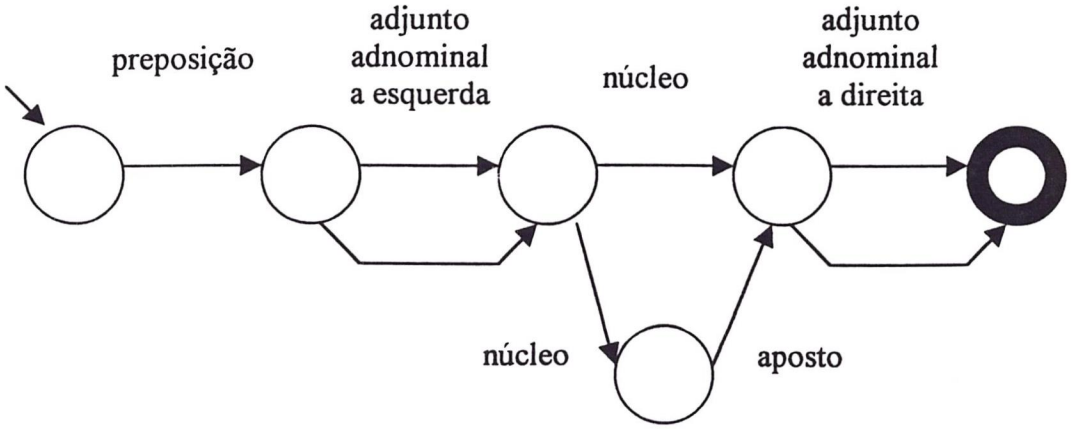



Figura 4 - ATN para Adjunto Adverbial

## 3. Aprimoramento do ReGra

O ReGra realiza o *parsing* de uma sentença de forma determinística, isto é, assim que se verifica a aplicabilidade de uma regra gramatical, ela é considerada integralmente pelo *parser*, até que a ATN correspondente seja completamente percorrida (caso de análise bem sucedida) ou até que o *parser* detecte um erro de natureza sintática e não consiga prosseguir com a análise (casos de detecção de incorreções gramaticais ou de inadequação do comportamento do ReGra). A opção pelo determinismo durante a análise sintática se deveu à necessidade de eficiência do sistema, que deve fornecer respostas em tempo real, enquanto o usuário está digitando seu texto.

Os problemas decorrentes dessa estratégia são vários, dentre os quais destacamos a ocorrência da ambigüidade lexical, que pode guiar o *parsing* para a aceitação de regras inadequadas, fato este que evidencia a ocorrência de falsos erros e falsos acertos, conforme já citados na Seção 1. O aprimoramento do ReGra visa a desambigüização lexical e, conseqüentemente, a melhora do *parsing* e do *checking* posterior, pela análise de viabilidade de adoção de novos métodos de análise ou revisão gramatical. Considera-se aqui, além dos métodos analíticos no nível da sintaxe ou semântica, também métodos estatísticos.

Para este aprimoramento, primeiramente foi realizado um procedimento analítico, para reconhecimento das falhas do ReGra. Em seguida, procedeu-se ao estudo de modelos adequados para se tratar alguns problemas encontrados, dentre os quais ilustramos, a seguir, os problemas de análise da forma "SE" no português.

### 3.1. Análise do "SE"

Devido à constatação de que o ReGra se comporta inadequadamente quando há a ocorrência da forma "SE" em sentenças sob análise (vide discussão em Martins et al., 1999), este trabalho consistiu em averiguar, do ponto de vista computacional, as ocorrências de análise do *parser*, ou seja, como o *parser* se comportava ante construções envolvendo tal forma, já que esta possui várias classificações na língua portuguesa, a saber:

- Substantivo;
- Pronome;
- Parte integrante do verbo;
- Partícula de realce;
- Partícula apassivadora;
- Índice de indeterminação do sujeito;
- Conjunção.

Para construções dessa natureza, o problema de análise do ReGra se agrava à medida que o "SE", dependendo de sua classificação, pode ou não assumir função sintática na sentença. Os principais problemas encontrados, neste caso, são:

- Especificações lexicais erradas como, por exemplo, na sentença "Estão-se lá". O ReGra aceita esta forma como correta, classificando o SE como parte integrante do verbo, pois o verbo "estar" está classificado como pronominal no léxico.
- Eleição indevida de regra gramatical durante o *parsing*. Devido ao processamento determinístico do modelo de ATNs e à própria ambigüidade lexical, há casos em que a escolha de uma das possíveis regras de *parsing* exclui considerações posteriores de outras regras desse elenco, para o mesmo contexto de análise. Nesse caso, o ReGra é incapaz de detectar a inadequação da escolha realizada, assumindo que o *parsing* pode ser concluído satisfatoriamente, apesar das inconsistências geradas.
- Ausência de regras adequadas para o contexto em foco. A incompletude do elenco de regras gramaticais que compõem o ReGra é um problema do próprio processamento da língua portuguesa em um contexto totalmente aberto e remete à dificuldade clássica de representação do conhecimento da própria Inteligência Artificial.

De forma geral, os problemas identificados acima são os responsáveis pela maior parte dos diagnósticos inadequados do ReGra durante o *parsing* e o *checking*, no que diz respeito à ocorrência dos falsos erros e falsos acertos. Apresentamos, a seguir, uma proposta de alteração do ReGra, para minimizar tais ocorrências.

## 3.2. Uma Proposta para a Inserção e Manipulação de Informação Semântica no ReGra

Buscando, ao mesmo tempo, determinar as razões pelas quais o ReGra se comporta inadequadamente, como no exemplo acima, e o modo pelo qual será possível aprimorar esse comportamento, no sentido de minimizar a ocorrência de inadequações de diagnóstico do ReGra, foi realizado um amplo estudo sobre diversas teorias semânticas (Abrahão e Lima, 1996; Katz and Fodor, 1963; Marcus, 1980; Marrafa, 1996; Pustejovsky and Boguraev, 1996; Silva e Lima, 1996) e métodos computacionais (Beesley and Grefenstette, 1996; Church and Mercer, 1993; Lucchesi and Kowaltowski, 1993; Pacheco et al., 1996).

A partir desse estudo, foi sugerido um modelo de processamento para a inclusão de processamento semântico no ReGra, a fim de resolver alguns casos de ambigüidade lexical. Essa sugestão será incorporada a um protótipo de um "novo" ReGra, para uma avaliação preliminar. Caso esta seja satisfatória, será avaliada a possibilidade de alteração do próprio sistema ReGra atual.

Do ponto de vista de modelagem lingüística, a proposta se baseia no modelo de subcategorização da Gramática Léxico-Funcional (Bresnan, 1982), segundo o qual o

verbo é o responsável por determinar seus argumentos através de traços semânticos. Cinco fases do projeto e desenvolvimento do protótipo foram delineadas, como segue:

- Coleta e análise de sentenças que contenham verbos de sentido. Esta etapa visa obter subsídios para a seleção de alguns casos passíveis de tratamento semântico automático, no contexto do ReGra;
- Determinação dos traços argumentais de cada verbo delineado na etapa anterior, incluindo, principalmente, os aspectos semânticos pertinentes a cada um deles;
- Determinação de uma distribuição ontológica dos itens lexicais;
- Atualização do léxico atual, para incorporar as informações semânticas.

A seleção dos casos verbais de interesse se baseia, primeiramente, em medidas estatísticas. Está sendo utilizado o aplicativo WordSmith (http://www.liv.ac.uk/~ms2928/wordsmit.html) para obter uma distribuição de freqüência dos verbos de sentido, assim como a correspondente distribuição contextual dos mesmos.

A partir dessa análise empírica, serão determinados "exemplos de teste", para um estudo preliminar do protótipo. Adotou-se, aqui, um método de desenvolvimento "nuclear", segundo o qual somente alguns casos de representação gramatical (específicos) serão tratados. A partir da especificação lingüística para esses casos, será especificada a configuração necessária do protótipo. Uma vez que este esteja funcionando satisfatoriamente para tais casos, a idéia é repetir tal estratégia para expandir gradativamente a potencialidade de representação do protótipo, de modo que ele venha a contemplar as mesmas construções gramaticais e o mesmo conhecimento lingüístico que compõem a versão do ReGra.

Do ponto de vista computacional, a proposta de desenvolvimento do protótipo para testes de viabilidade e eficiência do novo sistema parece caminhar no sentido de se implementar *demons* (programas disparadores de outras rotinas) (Tanenbaum, 1995) em certos pontos das ATNs já definidas no *parser* do ReGra. Por exemplo, considerando-se a oração simples (S-V-O) "o *menino comprou o doce*", a análise pode ser realizada com a ativação de um *demon* após o reconhecimento do verbo principal e após o reconhecimento de um marcador de fim de sentença. Desse modo, o primeiro *demon* restringiria o número de possíveis sentidos do verbo de acordo com o contexto, através do preenchimento de sua estrutura argumental em função das informações do sujeito, permitindo uma seleção mais apurada das regras sintáticas aplicáveis. Em seguida, ao fim do reconhecimento da sentença, o segundo *demon* completaria a especificação argumental do verbo com as informações semânticas relativas ao objeto da oração. A Figura 5 mostra um exemplo do posicionamento dos *demons* na análise sintática de sentenças dessa natureza, isto é, do tipo sujeito-verbo-objeto.

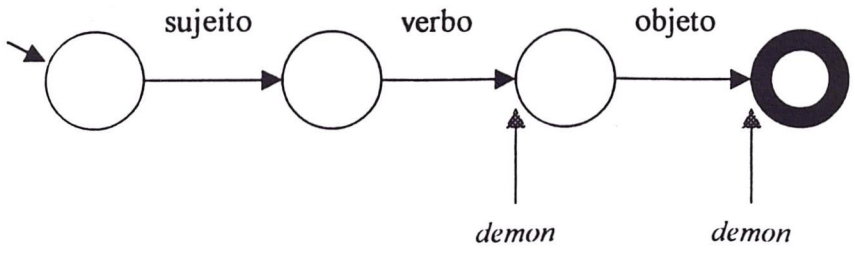



Figura 5 - Inserção de *Demons* na ATN

*Demons* diversos, para verificação e complementação semântica, seriam introduzidos nas ATNs do ReGra sempre que necessário, visando aumentar a probabilidade de se obter a representação correta do significado da oração sob análise. Deste modo, um possível procedimento para o *parsing* realizado com o auxílio de *demons* poderia ser especificado de maneira similar à sugerida por (Viegas and Raskin, 1998) e (Roche and Schabes, 1997), como segue (consideramos, aqui, como exemplo, orações S-V-O):

- Informações sobre cada componente sentencial são reservadas para processamento posterior enquanto não se encontra o verbo principal da oração;
- Ao se recuperar as informações dicionarizadas do verbo, um *demon* será responsável por verificar se os componentes já processados, assim como os componentes seguintes, satisfazem a estrutura argumental do verbo, guiando o processamento para a determinação das possíveis regras de *parsing* aplicáveis e das acepções lexicais apropriadas.

Portanto, para a correção gramatical de uma sentença, o *parsing* deve tanto verificar as formações sintáticas da sentença como sua correta correspondência com as informações semânticas.

O incremento no processamento do ReGra através do uso de *demons* pode acarretar, entretanto, um aumento do tempo de resposta do mesmo para a detecção e correção de um erro. Para equilibrar ou minimizar tal problema, as seguintes técnicas podem ser adotadas:

- Assim como o processamento sintático, o processamento semântico também seria realizado de forma determinística para maior eficiência do sistema;
- O processamento semântico poderia ser interrompido e ignorado sempre que um tempo limite de espera de resposta expirasse ou não houvesse correspondência entre as regras sintáticas e semânticas. Desta forma, o *parsing* verificaria somente a formação sintática da sentença.

### 3.3. Investigação de Alguns Modelos de *Parsing*

Do ponto de vista da implementação, também buscou-se avaliar alguns modelos de *parsing* existentes na literatura, a fim de averiguar as possibilidades de alteração ou adoção de mecanismos que pudessem contemplar a resolução da ambigüidade categorial (p.ex., (Sells, Shieber and Wasow, 1991)), cujas fundamentações remetem às gramáticas normativas (ou de estrutura de frase), como as livres de contexto, sensíveis ao contexto, ATNs, de cláusulas definidas (DCGs), categoriais (CGs), léxico-funcionais (LFGs), de dependência (DGs), de árvores de adjunção (TAGs), ou às gramáticas de dependência ou de casos. No entanto, os modelos investigados não se aplicam necessariamente aos requisitos, objetivos e ao próprio contexto do ReGra, de revisão gramatical e, de modo geral, recaem nos problemas complexos de PLN apresentados no ReGra, na maioria das vezes procurando resolver somente alguns deles (em *toy systems*), não servindo como base para a aplicação real no domínio aberto em foco.

## 4. Conclusões e Perspectivas

O código do ReGra foi implementado de modo eficiente e, atualmente, consiste no revisor gramatical mais apurado existente para a língua portuguesa. Seu código é

sucinto, havendo sido otimizado ao longo de suas diferentes versões. Entretanto, a ambigüidade lexical é responsável pela necessidade de refinamento, para evitar a maior parte dos casos de análise incoerente, em termos gramaticais. Para solucionar (ou minimizar) tal problema, uma proposta de processamento semântico foi apresentada, aparentando ser viável para sentenças com estruturas simples. Contudo, muito trabalho ainda deve ser despendido para a melhor determinação dos casos a serem tratados.

## Referências bibliográficas


Abrahão, P. R. C. e Lima, V. L. S. (1996). *Um Estudo Preliminar de Metodologias de Análise Semântica da Linguagem Natural.* II Encontro para o Processamento Computacional de Português Escrito e Falado. Centro Federal de Educação Tecnológica do Paraná , Paraná. Outubro.

Beesley, K. R. and Grefenstette, G. (1996). *Finite-State Analysis of Written Portuguese.* II Encontro para o Processamento Computacional de Português Escrito e Falado. Centro Federal de Educação Tecnológica do Paraná, Paraná. Outubro.

Bresnan, J. (1982). *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass.

Church, K. W. and Mercer R. L. (1993). *Introduction to the Special Issue on Computational Linguistics Using Large Corpora.* Association for Computational Linguistics.

Roche, E. and Schabes, Y. (1997). *Finite-State Language Processing.* The MIT Press. Cambridge, Massachusetts.

Katz, J. J. and Fodor J. A. (1963). *The Structure of a Semantic Theory.* Linguistic Society of America, Arlington, E.U.A.

Lucchesi C. L. and Kowaltowski T. (1993). *Applications of Finite Automata Representing Large Vocabularies.* John Wiley & Sons Ltda.

Marcus, M. P. (1980). *A Theory of Syntactic Recognition for Natural Language.* The MIT Press. Cambridge, Massachusetts and London, England.

Marrafa, P. (1996). *Representação das Formas Predicativas Verbais do Português numa Base de Conhecimento Lexical.* II Encontro para o Processamento Computacional de Português Escrito e Falado. Centro Federal de Educação Tecnológica do Paraná , Paraná. Outubro.

Martins, R.T.; Rino, L.H.M.; Montilha, G. e Nunes, M.G.V. (1999). *Dos modelos de resolução da ambigüidade categorial: O problema do SE.* IV Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR'99). Universidade de Évora, Portugal. Setembro.

Pacheco, H. C. F.; Dillinger, M. e Carvalho, M. L. B. (1996). *Uma Nova Abordagem para a Análise Sintática do Português.* II Encontro para o Processamento Computacional de Português Escrito e Falado. Centro Federal de Educação Tecnológica do Paraná , Paraná. Outubro.

Penteado, R. D. (1996). *An Overall Process Based on Fusion to Reverse Engineer Legacy Code.*

Pustejovsky, J. and Boguraev, B. (1996). *Lexical Semantics. The Problem of Polysemy.* Clarendon Press, Oxford.

Sells, P.; Shieber, S. M. e Wasow, T. (1991). *Foundational Issues in Natural Language Processing*. The MIT Press. Cambridge, Massachusetts.

Silva, J. L. T. e Lima, V. L. S. (1996). *Algumas Considerações sobre Resolução da Ambigüidade Léxica no Processamento da Linguagem Natural*. II Encontro para o Processamento Computacional de Português Escrito e Falado. Centro Federal de Educação Tecnológica do Paraná , Paraná. Outubro.

Tanenbaum, A. S. (1995). *Sistemas Operacionais Modernos*. Prentice-Hall do Brasil Ltda. Rio de Janeiro – RJ.

Viegas, E. and Raskin, V. (1998). *Computational Semantic Lexical Acquisition. Methodology and Guidelines*. New Mexico State University. Las Cruces, USA.

Woods, W.A. Transition Network Grammars for Natural Language Analysis. *CACM* 13, pp. 591-606.